NAMA PELAJAR	:	**RACHAEL VOO SZE YING**
NO. MATRIK	:	**WEK020213**
KOD KURSUS	:	**WXES 3182**
TAJUK KURSUS	:	**PROJEK ILMIAH TAHAP AKHIR II**
TAJUK PROJEK	:	**DIGITAL GAME-BASED LEARNING: C++**
NAMA PENYELIA	:	**PN. SITI HAFIZAH BTE ABDUL HAMID**
NAMA MODERATOR	:	**P.M. RAJA NOOR AINON ZABARIAH RAJA ZAINAL ABIDIN**

ABSTRACT

Game-based learning has been recognized as an important alternative to traditional method of teaching. It can help both adults and children in learning new concepts, gaining expertise and practicing knowledge. Though game-based learning is mostly applied in children's education, it can be helpful for adult vocational and university-level learning.

Digital Game-based Learning: C++, which I prefer to refer to as Digital Game-based C++ Learning System is a web-based system that promotes learning C++ through games. It is the first of its kind whereby no programming language is to be learned through games. So, basically this system's objective to provide aid in learning basic concepts of procedural programming, enhance learner's understanding of procedural programming and motivate learner to be more interested in learning C++ programming language.

It targets especially students or learners who are currently learning programming using C++ language and they felt that they need extra help in understanding the concepts of C++ programming.

ACKNOWLEDGEMENT

This thesis report is brought to completion with the generous help of a number of persons. These individuals stand out in a special way: Mrs. Siti Hafizah, supervisor for WXES 3181 subject, who has kindly guide me through the report and giving me advice to make sure everything goes smooth; Associate Professor Raja Ainon, my moderator.

Not to forget also, my friends who have helped in various ways, by giving me their opinions, answers to unknown questions and etc.

No acknowledgement would be complete without thanking God, from Whom all blessings flow.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1:

INTRODUCTION

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION OF THE PROJECT

Games have always been part of our lives as humans. We play games for amusement, we play games for fun, we play games for leisure and we even play games just for the sake of playing. It has never crossed our minds that sometimes we play games to learn. Look back at the time when we were toddlers. Remember that our mommies bought those cute, colourful building blocks for us? From those small building blocks, we build our imaginary houses, cars and airplanes. It has never occurred to us then we are learning to be creative, learning to put those blocks to use.

1.1.1 Games and Education

Somehow or rather, it took us quite a while to put games into use in classrooms. We always have bias and negative perspective that games are only for entertainment. Well, if you still have doubts that games are only for that purpose, it is time that you erase that mind set of yours and be open to new changes.

A well-known writer has once said, "The fundamental of the school system is its failure to motivate youth of the country to want to learn". When I reflect on the quote, it is undeniably true that learning without motivation is futile. People need to be motivated to learn. The best learning method is a method where the learner to able to build on previous knowledge that he or she has. This method of learning is sometimes refers to as scaffolding. The more concrete the initial experience was, the more likely the learner is

able to understand what was being taught. The scaffolding method can be implemented through multimedia games.

Through such games, learner is motivated to try new things. One of the key factors in learning is again, motivation. Without motivation, there will be no interest. No interest means learning by force. Learning by force is no better than not learning at all.

Besides that, some people learn better visually and some kinesthetically, that means a combination of visual and action. Games are able to provide that. It is a known fact that people retain more knowledge when they are actually experiencing what they are learning.

With the introduction of multimedia games, education is taken to another level. Learning is no longer boring sessions in the classroom with teachers. Students now have a more exciting way to learn – that is through a combination of text, sound, video, graphic and animations called game-based learning. Multimedia games are attractive and engaging and most importantly, it is FUN!

Game-based learning has already been adopted for children's education. It is available mostly in the form of CD-ROMs sold in the market. It is proven that there are improvements on children's learning when game-based learning is adopted. Subsequently, if games can be used for children's learning, it can be adopted for adult education as well.

1.1.2 Problem Definition

Taking Faculty of Science Computers and Information Technology (FCSIT) of University of Malaya as an example, up till today, FCSIT has no supplement learning tools for students in the subject of C++ programming, other than written notes. Students who are not interested in reading notes may have some difficulties in learning C++ programming when all the educational institution provides are text-based notes.

In the market today, learning aids for C++ programming consists only of reference books, electronic-books that come in the form of CD-ROMs and not to forget, notes and tutorials on the Internet. All these however, are all text-based. Student, who needs greater motivation than reading, may find it hard to catch up in programming. To increase the student's motivation in learning, a new learning tool must be adopted. The learning tool could be either web-based or CD-ROM based that replaces traditional techniques of teaching with a better, interesting and interactive ways of learning.

Digital Game-based C++ Learning System is a proposed system that incorporates such learning techniques stated above. With the development of this system, it will allow extensive interactivity between the user and the system which in return, may results in better performance due to improved motivation, and greater understanding due to greater initiative to learn.

1.1.3 Project Overview

All these while, programming languages such as C, C++, and Java are taught through reference books and lab exercises. Regardless of which institution that provides these courses or the learner self-taught themselves, traditional method of teaching and learning is still adopted. This proposed system, **Digital Game-based C++ Learning System** is to provide an alternative way of learning by build a web-based system that integrates games especially to learn the C++ programming language.

The proposed system has 6 main modules:

1) Sign-In
    - Sign-In module is a module where registered (existing) user is required to login into the system to access contents of the system
    - The Sign-In module is also equipped with a register function that enables new users to register themselves into the system.
2) Notes
    - This module provides short notes related to C++ programming. Notes can be divided into 5 main chapters. In each chapter, information provided is short and easy to digest.
3) Tutorials
    - It provides extra exercises for the registered user on topics related to C++ programming.
4) Games

- This is the main content of the system whereby this module provide games to aid in learning C++ programming language

5) Code Dump

- This is a function that enable user to download C++ sample source codes as a reference.

6) Help

- This is a function where help is provided for first time user and user who are unsure what to do with the system

7) Maintenance

- This module is to maintain and keep the website up-to-date with current notes, tutorials, games, and source codes. This module is specifically for the administrator.

1.2 PROJECT RATIONALISM

Learning C++ programming language poses great learning difficulties especially for learners who are bored by just reading reference books and at the same time, dislike doing programming exercises. Although some may have the initiative to learn, somehow or rather, in the middle of the learning process, these learners may be hampered by all sorts of terms and concepts they have to remember and apply. Thus, making the learning process easier for these learners and keeping them motivated at the same proves to be a tough task for the educators. One viable solution is learning through games or simulation.

Learning through games brings uncountable benefits because it can replace traditional techniques of teaching with innovative forms of learning that is interesting, and at the same time, motivating and encouraging life-long learning.

Currently, there are lots of C++ learning package available either online or in CD-ROM titles but all of these packages are merely books available in digital form. Due to the unavailability of medium that promotes learning C++ programming language through games in the market today, this project is certainly feasible and suitable to be developed.

1.3 PROJECT OBJECTIVES

Objectives of the project are:

- Provides aid in learning basic concepts of procedural programming in C++ programming language
- Enhance learner's understanding of procedural programming in C++ by playing games
- Motivate learner to be more interested in learning the C++ programming language and subsequently, other programming languages

1.4 PROJECT SCOPE

This project only covers C++ procedural programming. It does not include object-oriented programming chapters that C++, as an object-oriented programming language offers.

Chapters that are included in this project are as follows:

- Basics in procedural programming
- Control Structures
- Functions
- Arrays
- Pointers

Chapters that are NOT included are (Object-oriented programming concepts):

- Classes
- Overloading
- Inheritance
- Polymorphism
- Exception handling

1.5 TARGETED USER

The targeted users of this software produced are those who are learning to program using C++ but needs extra help in understanding the concepts of programming.

1.6 EXPECTED OUTCOME

The system is able to provide the user with:

- A user-friendly website that will give the learner the freedom to explore and understand the chapters they are weak in

- Help student improve their performance and get a better idea on C++ procedural programming
- Games that promotes more clarification on C++ programming theoretically and practically.

1.7 PROJECT SCHEDULE

The Figure 1.1 below shows the schedule of each stage of the project.

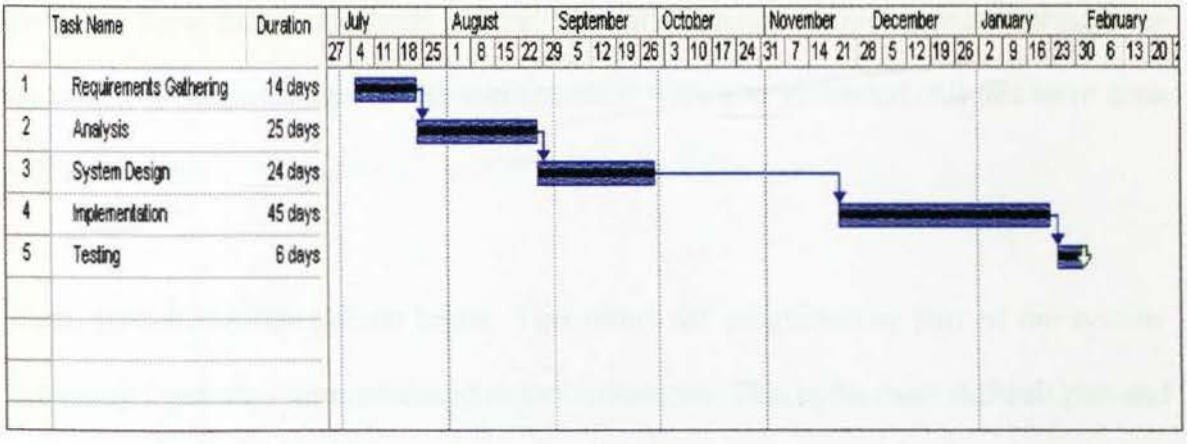


Figure 1.1: Schedule of the project

The project started off in July 4th, 2004 with requirements gathering. This is where facts on existing and/or related system were gathered and compiled. Other than that, hardware and software available in the market are also researched so that suitable tools can be chosen for design and development. Requirements gathering also look into what the users expectations and requirements from the project. This was done by handing out questionnaires to various respondents from FSKTM of University of Malaya.

Requirements gathering took around 14 days to complete and it was followed by analysis of the data gathered. In this stage, the users' functional and non-functional requirements were generated. Tools to be used for implementation were also decided. On the whole, analysis took 25 days.

Analysis was followed by system design. Functional and non-functional requirements were put into design. This was the preliminary stage to system implementation. Class diagrams, flowcharts, statecharts and etc. were all produced in this stage to represent the upcoming system. Designs of the user interface were also generated. All this were done within 24 days.

Then, system implementation began. This refers the programming part of the system. Games and websites were produced as the milestones. This is the most difficult part and took the longest time to complete.

After 45 days in the implementation stage, system testing began. This stage is to ensure that the whole project run smoothly and correctly without any errors. Any errors were eliminated during this stage.

1.8 CHAPTER SUMMARY

This chapter is mainly focused on the proposed system. It includes discussion of problems with today's method of learning, proposed system's objectives, scope and expected outcome. The remainder of the report is organized in the following manner:

Chapter 2: Literature Review

-This chapter evaluates and summarizes works and researches related to my area of research that is digital game-based learning of C++ programming language. This includes not only existing and related system and software but also on tools, application software and databases suitable for developing the system.

Chapter 3: Methodology

-This chapter looks on at system development processes and methodologies available to develop the system and techniques used to discover requirements for the system.

Chapter 4: System Analysis

-This chapter is about analysis of requirements elicited from the previous chapter whether they are feasible or infeasible. Functional and non-functional requirements that are crucial and related are identified.

Chapter 5: System Design

-This chapter puts the functional and non-functional requirements into design.

Chapter 6: System Implementation and System Programming

-This chapter looks at how the design is being implemented to create the system

Chapter 7: Testing and Evaluation

-This chapter looks at the testing employed to ensure that the system works properly and users' evaluation on the system

Chapter 8: Conclusion

-This chapter summarizes the whole project. It also looks at problems faced during the development, solutions to these problems, project's strength and limitations and lastly the enhancement that could be done in near future.

CHAPTER 2:

LITERATURE

REVIEW

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

When the word "literature" is used, we often think it as written works, such as novels, plays and poetries by famous authors, for instance, William Shakespeare, Emma Austen and Thomas Hardy. However, in the context of literature review, it means something different. "Literature" connotes works consulted for the purpose of comprehending and investigating a research problem. In this case, the research problem is anything that relates to digital game-based learning of C++ programming language.

2.1.1 What is literature review?

Literature review has a diversity of definitions. It is said that literature review is an evaluative report of a selected area of study (McKillup *et al,* 2000). Another would say it is a critical look at the existing research that is significant to the field of study. Yet, another thinks that it is an account of what has been published on a topic by scholars and researchers (Taylor & Procter, 2004).

All are these definitions are feasible due to the nature of the literature review itself. Literature review IS an evaluative report, IS a critical look and IS an account of works related to the field of your research.

According to Bruce (Bruce, 1990), a review of relevant literature is nearly always a standard chapter of a thesis or dissertation. The review forms an important chapter in a

thesis where its purpose is to provide the background to and justification for the research undertaken.

Though there are numerous different definitions, there are too some similarities in their definitions. First, literature review is NOT a summary. Yes, relevant researches are summarized but it is also crucial to *evaluate* all works, show *relationships* between different works and show *how it relates* to the area of your work.

2.1.2 Why write a literature review?

The basic purposes of writing a review are:

- Provide a context for your research (position your work relative to other existing works) (Scown *et al,* 2000)
- Justify and ensure that research has not been done before - avoid reinventing the wheel (Scown *et al,* 2000; Macaulay, 2004)
- Carry on from where others have already reached (Macauley, 2004)
- Identify other people working in the same field (Macauley, 2004)
- Identify flaws, opposing views and outline gaps in previous research (Scown *et al,* 2000; Macauley, 2004)
- Increase knowledge in the subject area (Macauley, 2004)
- Help refine, refocus or change the topic (Scown *et al*, 2000)

2.2 INTERACTIVE MULTIMEDIA

The concept of multimedia was introduced decades ago. Technology to bring multimedia to the desktop made its first step into the world in the late 1970s when the microprocessors were born which spurred the emergence of personal computers. Desktops used then have evolved from single-purpose computers to sophisticated, multi-purpose computers that help to get jobs done, provide information and entertainment. Timeline of technology behind the creation of multimedia computers is shown in Figure 2.1. It is adapted from Hillman's "Multimedia Technology & Application" book (Hillman, 1998):

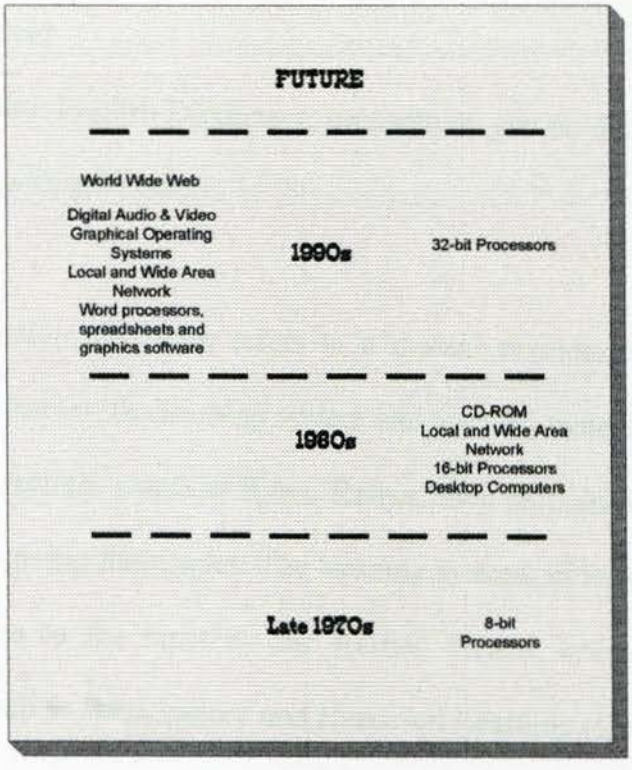


Figure 2.1: Timeline of Technology Behind the Creation of Multimedia Computers

Historically, multimedia merges only text and graphics. However, in between the 1980s and 1990s, the concept of multimedia took a new turning when capabilities of satellites, computers, audio and video are converged to create a fresh type of media with enormous potential. These powerful combinations of technologies have the ability to provide intensive and enhanced learning potential which focus on to the specific needs of individual users. Technologies as such are known as interactive multimedia.

To understand interactive multimedia, it is wise to start off with the definition of a few terms.

2.2.1 What is Interactive?

According to Collins Gem English Dictionary, interactivity can be defined as acting on or in close relationship with each other.

In the context of education, this term refers to a mutual relationship that promotes interaction between the learner, the learning system and learning material. In a sense the learner does more than just sit, watch or listen. Rather, they participate in the learning system and material by giving their input. The learning system, in return, will produce some output in response to the input by the learner. Hence, interactive is usually associated with terms such as Participatory and Hands-on Learning (Centurion Systems Corporated, 1997).

In the context of technology, interactive technology is any technology that is able to accept user inputs and produce outputs conditionally and non-linearly (Locatis, *unknown year)*.

Other than that interactivity also refers to the ability of the user of a system to control sequences of events in an application, especially in navigation of presentation and influence of the content seen onscreen. From this definition, interactivity can be in the form of navigational buttons, hypertext and hypermedia.

Interactivity, from all the definitions above can be deemed as the key feature of all multimedia. The user is given some influence over the access to information and also a certain degree of control over the outcomes of the system. In other words, it allows the content of the system to be presented in a non-linear way, determined by the user himself.

2.2.2 What is Multimedia?

Multimedia is hard to pin down to a rigid and accurate definition. It means differently when applied to different people in different situation. Among the various definitions are:

- "Multimedia technology is any technology capable of presenting combinations of text, sounds, pictures, animations, and/or full motion video" (Locatis, *unknown year*).

- "Multimedia is an umbrella term that has been coined to cover all of the synergistic users of text, voice, music, video, graphic and other forms of data to enhance the computer's role as a communication device" (Dillon & Leonard, 1998).

- "Multimedia is a seamless integration of data, text, images of all kinds and sound within a single, digital environment" (Feldman, 1994).

- "Multimedia can be defined as a computer-based interactive communications process that incorporates text, graphics, sound, animation and video" (Shuman, 1998).

In a nutshell, multimedia is an integration of multiple media elements, namely, audio, video, graphics, text and animation, that conveys information.

2.2.2.1 Multimedia Fundamentals

Multimedia consists of 5 main elements:

- Text
- Graphics (images)
- Audio (sound)
- Video
- Animation

Table (Table 2.1) below shows the description of each multimedia element:

Table 2.1: Description of multimedia elements

ELEMENTS	DESCRIPTION

Text	• It is the usual way which information is dispensed
	• Text can vary depending on the function of the application
	• Textual information must exist in harmony with other parts of elements and design processes
	• Text can exist in different fonts. There are 2 groups of fonts
	  ○ Serif
	  ○ Sans-Serif

Graphic	• Illustrations or other figures to represent something in a pictorial way and identify them visually • It may be used when complex situation arises where text usage may cause ambiguity • To use graphics in any application, it require knowledge, skills and expertise in creating and manipulating them • Different graphic packages create different types of files formats such as bitmaps, JPEGs, Tiffs and etc.

Audio	• Also known as sound to some.
	• It contributes to learning experience
	• It is the descriptive part of a particular object or simulation
	• Recording, editing, embedding and play-back of audio file requires in-depth knowledge of digital audio
	• There are 2 types of audio:
	  ○ MIDI
	  ○ Digitized sound

Video	• It is a seamless integration of moving images and synchronization of audio • Can make dramatic changes especially to a multimedia project
Animation	• It is an evolution of static pictures or graphics • It consists of series of pictures that are strung together and played at a certain rate per second to

	emulate movement • Using animation requires knowledge in 2D and 3D graphics and images manipulation techniques

2.2.3 What is Interactive Multimedia?

"Any sufficient advanced tech is undistinguishable from magic" Arthur C. Clarke

Interactive multimedia is any technology that consists of a combination of texts, sounds, pictures, animations and/or videos that conveys and presents information mutually between the learner, the learning system and the learning materials, whereby the learner provides inputs and in response, the learning system with the materials produce outputs conditionally and non-linearly.

Interactive multimedia is also known as a "hybrid technology". Why? This is because it integrates storage and retrieval capabilities of database technology with advanced tools for viewing and manipulating these materials (Bass, *unknown year*).

2.2.4 Interactive Multimedia in Educational Context

Why use multimedia in learning? Do we need multimedia to achieve educational objective?

In this era of technology proliferation, it is certainly sensible to try to include some aid into the existing educational system to help make education more effective. Multimedia has the capability to offer unique advantages into the education field.

First, multimedia, with its extraordinary storage and delivery capabilities, is a powerful and efficient source for acquiring and providing learning resources especially for schools and learning institutions that has difficulty in obtaining books and references. It allows educational institutions to access all kinds of inaccessible material and even put resources from the remotest locations from major research facilities at the fingertips of users.

Secondly, multimedia gives the user a control over materials collected and acquired. Interactive multimedia programs enable the user to manipulate these materials collected through linking, sorting, searching and annotating activities. Each of these activities can be made to reinforce and inculcate various intellectual skills.

Other than that, for some student, text alone is not sufficient for them to understand what has been taught. Imagine learning to tie shoelaces with a manual. Text alone simply makes he/she more confused and disallow them to get the picture to tie shoelaces properly. According to a research (Menn, 1993), it is said people retain 10% of what they see, 20% of what they hear, 30% of what they hear and see, 50% of what they see and explain at the same time and a booming 80% of what they see, hear and do simultaneously. Isn't that amazing?! It is with this fact that multimedia should be immediately be implemented and embedded into the education system today.

Referring to a computer-based training research, multimedia training offers 60% of the learning curve factor, 25% to 50% of higher content retention, 56% higher learning

gains and 20% to 40% of lesser variances in quality of delivery. From this research, can be concluded that multimedia is an effective tool and certainly a catalyst of education.

2.2.5 What kind of Interactive Multimedia Available?

There are a lot of different ways to structure the different kinds of multimedia that are currently available in the market. Randall Bass (Bass, *unknown year*) groups multimedia into five distinct categories. These five categories are shown in the table (Table 2.2) below:

Table 2.2: Categories of Interactive Multimedia

CATEGORIES	DESCRIPTION

Educational Packages : General & Comprehensive History	• Courseware packages with broad coverage
	  ○ Chronology
	  ○ Comprehensive content
	• Contain extensive print, graphics, audio and video materials
	  ○ Create rich information environment that serve as a supplementary curriculum to a diversity of historical and cultural courses
	  ○ Informative, standalone research environment for teachers, scholars and public

Educational Packages : Focused Topics & Collection	• Similar to educational packages of general & comprehensive history
	• Particular themes or disciplinary focus
	• Function as
	  ○ Teaching tools
	  ○ Reference tools

Historical Simulation	• Educational packages designed as computer-based simulations • Exploit the capabilities of computing technology to calculate variables and

	simulate possible scenarios • Re-centers the focus of classroom teaching away from the teacher • Focus on collaborative group work among student • Focus around contingent problems of decision making and hypothesis testing
Electronic Texts	• Literary, historical and cultural texts in electronic form • Loaded into software with search and analysis capabilities (and multimedia linking) • There are different kinds of electronic texts o Text as electronic "books" o Text in multimedia packages o "WordCruncher" software that is neither a multimedia or hypertext • Aids in teaching and research activities
Interactive Educational Reference Tools	• Function as reference tools in the context of education and classroom teaching or library situations • Only a portion of these tools are interactive multimedia

These categories are not mutually exclusive and it is natural that multimedia programs are a combination of 1 or more categories mentioned in the table.

2.2.6 Advantages of Interactive Multimedia for Learning

Interactive multimedia offers heaps of benefits in both teaching and learning, some of which are listed below:

- Caters to different learning styles

  Information is organized and presented with a range of media. The range of media forms support a variety of learning styles and allows the learner to make

choices of content to meet their needs. It is a known fact that some people learn best from text, while others may require graphical or aural presentation.

- Provides consistent delivery, feedback and evaluation

  Interactive multimedia programs deliver the same content to everyone. No variation is seen in content delivery, feedback and evaluation. These features enhance the equity and inclusiveness of training and eliminating any sense of discrimination that an individual may feel in a group situation. Learners too, can choose to assess themselves when they are totally confident in the subject matter without pressure from assessment and evaluation by other people

- Self-directed and self-paced

  Interactive multimedia programs allow and encourage learners to work by themselves at their own time. It gives the learner control over the range of material used and allows the learner to define his own preferred learning path and styles so that he is able to make meaning from the information presented to them. It also aids learner to centralize on the learning process rather than having the need to focus and be dependent on a facilitator or teacher.

  In other words, it gives the learner the flexibility of "anytime, anywhere" learning and greater individualization in learning.

- Reduces inaccessibility of learners

  Interactive multimedia allows learners to easily attend to work and education at a moment's notice. Learners are not required to go to a specific facility or location just to facilitate learning. They have the flexibility to access learning information

anywhere, provided that the place they are in is equipped with proper and adequate equipment for online learning.

- Overcomes time constraints of classroom delivery

  Classroom learning usually has a schedule. Both teacher and learner are pressured to assimilate large amount of information in a short time frame. This often leads to "spoon feeding" and minimizes real learning and individual assessment.

- Makes learning gradual

  Learning over time is something that an interactive multimedia is able to facilitate. It provides opportunities for reflection and application. Furthermore, it is critical to change practice in the long term. It also supports gradual involvement with the learning material. Components of work can be undertaken at times appropriate to progress. Sequence of learning is determined by the learner himself.

- Gains and hold attention

Combination of various media makes people interested in what they are learning. Besides that, it engages multiple senses such as read, see, hear and actively manipulating materials. As the old says "a picture is worth a thousand worth". Communication research shows that the combination of communication modes, aural and visual, offers greater understanding and retention of information. As an educator enthusiastically put it:

> *As humans, we seem hard-wired for multiple input. Consider that we remember only about 10% of what we read; 20%, if we hear it; 30%, if*

we can see visuals related to what we're hearing; 50%, if we watch someone do something while explaining it; but almost 90%, if we do the job ourselves – if only as a simulation. In other words, interactive multimedia – properly developed and properly implemented – could revolutionize education. (Menn, 1993).

2.2.7 Disadvantages of Interactive Multimedia for Learning

When there are advantages, there are too disadvantage. The drawbacks of interactive multimedia in the context of learning are:

- Computer Literacy

  Not all potential learners are confident users of computers. They may not feel comfortable in using a computer. This adversely causes learning curve where the learner has to learn how to operate a computer first before using the computer to facilitate learning. At the same time, their learning initiative might be dampened by the fact that they are not confident enough to handle a computer.

- Equipment Reliability

  Computer hardware and software has become increasingly stable and reliable. Consistent and adequate program testing during product development may help to remove all technical glitches.

- Content Updates

  Content of the learning package has to be updated periodically to ensure information provided is up-to-date.

2.3 DIGITAL GAME-BASED LEARNING

2.3.1 What is a game?

A game can be defined as any activity for amusement or pleasure. For example, a game could consist of activities such as dancing, singing, skipping, playing with toy trains and etc. This definition is a result of extensive research by Johan Huizinga and Georg Jünger in the year 1938 and 1959 respectively.

However, as the years go by, more and more people came up with ways to define game. Manfred Eigen and Ruthild Winkler both define game as a natural phenomenon: half necessity and half coincidence. Adornos, another researcher concluded that game is an art form.

According to Osborne and Rubenstein, games are description of strategic interaction that includes the constraints on the actions a player can take and the player's interests, but does not specify the actions the player does take.

As we can see, again there are no rigid descriptions or definitions for game. It all depends on individuals on their opinion or perception of the word "game".

2.3.1.1 Criteria for a game

All games must have:

- Common experience

Games should be able to bring people together regardless of gender, generation and race (Kramer, 2000). There are games which can be played alone (also known as solitaire games) and games which enable multiple players.

- Equality

  All players engaging in a game are equal and provided the same chance to win (Kramer, 2000). At any one time, any player is given the chance either winner or loser.

- Freedom

  Players engage themselves in a game on his freewill or freedom of choice. Playing games in not work nor commitment. Freedom is the basic to all games (Kramer, 2000).

- Activity

  "Playing means being active" (Kramer, 2004).

  Based on the phrase above, activity is the basic to all games. Activities that a player engages in depend on the game itself. Activities that could be undertaken include:

  - Spiritual area which involves thinking and combining, planning, making decisions, concentrating, training of mind, receiving knowledge and understanding the impact of systems (Kramer, 2000).
  - Emotional area which involves rules and accepting laws, learn how to work with others, learn how to lose, learn more about yourself and others and use fantasy and creativity (Kramer, 2000).

- Motor area which involves practicing skillfulness and practicing reactions (Kramer, 2000).

- Diving into the world of the games

  Player engaging in a game leaves behind reality and dives into the world of game. Game worlds are comparable to reality. Both worlds are substantiated by the same factors:

  - Law/rules
  - Chance
  - Own doing
  - Competition (achievement of the best)
  - Course and end are unknown
  - Same language and means of expression

  However, it is important to keep in mind that reality and game world is different. A game world is NOT an everyday world nor is reality a game world. Everything that happens in a game world stays in the game world.

2.3.2 What is learning?

Cambridge Advanced Learner's dictionary defined 'learning' as an activity to obtain knowledge or skill in a new subject or an activity. Learning applies to every level of human hierarchy. Learning can be on any level of education, kindergarten, elementary school, college or even the outside world, corporate.

2.3.2.1 Web-based Learning

There are lots of hype and confusion about what web-based learning is and how it impacted our lives. It is actually a structural material deployed and managed through computers that are linked to the Internet. One of the components of web-based learning is none other than e-learning.

2.3.2.1.1 What is e-Learning?

E-learning is a convergence of Internet and learning. Basically, it refers to an approach to facilitate and enhance learning by means of personal computers, CD-ROMS (Compact Disc-Read Only Memory) and Internet. It has, literally, brings the classroom to the doorstep of the learner itself. It gives the impression of learning to "anyone, anyplace, anytime".

2.3.2.1.2 Benefits of e-Learning

The benefits of the e-Learning are:

- "Anyone, anytime, anywhere" concept
    - Learning is meant for everyone everywhere around the world. It has no specific time frame for learning whereby people learn whenever they feel like it. In other words, it meant that people is given the freedom to choose when they want to learn and where they want to learn.
- Substantial cost and time saving

- E-learning is geographical independence. It eliminates the need for the learner to travel to specific locations just for education. Thus, saving time and cost of travel incurred in traditional classroom learning.

- Promotes innovative and interactive learning
  - Allow diversity of learning styles involving different students, and cultivates cognitive learning that leads to a better understanding and recall of knowledge and skills.
- Improved collaboration
  - Enable stronger and more meaningful collaboration on a one-to-one, one-to-many and many-to-one basis, involving both students and teachers.
- Career Oriented
  - Acquire skills required in the job market.
- Just-in-time access to timely information
  - Makes learning easy for busy or part-time learners. They can have access to interactive, self-paced, multimedia training, anytime and anywhere.
- Self-paced learning
  - Cultivates self-paced learning in learners where they choose to learn at a speed that suits them
- Allows learning to be broken down into smaller units
  - E-learning courses constituted of "Learning Objects" that allow learning to be dissected into discrete learning units. Compared to conventional learning, students are forced to take up complete courses regardless of which course components learners wanted to take up

- Uniformity of content
    - Instructional quality is consistent to all level of learners irrespective of location anywhere in the world.

2.3.2.1.3 Disadvantages of e-Learning

There are unfortunately drawbacks of e-Learning. They are:

- Set of different course associated with the course provided
    - Other than paying for the fees associated with the course, the learner still has to purchase and implement a web-based delivery platform if the learner does not have one
- Focuses on delivery on information rather than foster interaction between the learner and the instructor that leads to effective, normal learning
- Unable to identify individual student problem that might help to improve the performance of the learner

2.3.2.2 Web-based in Education

Learning will fall short of its objective and potential of educating young and old minds if it merely "repackages" current educational model into digital format. The basic idea of web-based learning is to dock the classroom the Internet in a most efficient way possible. This is done so that we, as learners will have the opportunity to change the way we think and we, also, as the provider of information, have different way of delivering learning solutions. In addition to that, with the incorporation of e-learning in the existing

education system, learner is provided with an interactive way of learning and indirectly, motivates them to learn more and more in the near future.

2.3.3 What is Digital Game-based Learning?

Digital game-based learning is a type of learning that uses the medium of computers to engage us in games that have learning properties.

2.3.4 Why Digital Game-based Learning?

Most of us have grown up to be a part of the digital generation. From pagers to cellular phones, simple board games to computer and video games, it is undeniable that in the days to come we, humans, will too be "digitalized".

If you stop by an entertainment arcade, it is by a hair's breadth surprising to find it jammed with people so engrossed in the games they are playing. On the other hand, if you happen to pass by a classroom, you'll notice the attention level and interest of students plummet far below than what you saw in the arcade.

In the olden days, people have fixed mindset on games. Games are for entertainment ONLY while learning is something that must be seriously taken into consideration. Now, such thoughts can be classified as "paleolitic". Digital games offer an irresistible opportunity to capture the imagination and instruct the player.

2.3.5 Do games work?

Numerous researches have shown that digital games are useful in the educational context. How useful are they?

- Improves basic literacy skills (Brownfield & Vik, 1983)
- Improves reasoning skill (Wood & Stewart, 1987)
- Improves understanding of physics (Ravenscroft, 2002)
- Serves as the alternate forms of testing (Kennedy *et al,* 1982)

Games inevitably also have indirect positive effects on players engaged in it.

- More concentration, more persistent, more absorbed, learned better and more creative than subjects who did not learn via digital games (Starbuck & Webster, 1991)
- More fun and effective
- Motivated to learn
- Enables learner to build on previous knowledge and allow them to apply what they have learned to new or even real-life situation

2.3.6 Why do games work?

According to Ahlers & Garris (Ahlers & Garris, *unpublished*), who have been using game-based learning extensively at the Navy's submarine labs have suggested games are effective for education due to:

- Opportunities that it offers for success that lead to a sense of purpose
- Stimulation of one's curiosity, which indirectly leads to fascination

- Simulation of danger that proves to be stimulating
- Social reinforcement which leads to a sense of accomplishment

Marc Prensky (Prensky, 2001) claims that games work because it provides engagement and interactivity.

2.3.7 Varieties of Digital Game-Based Learning

Marc Prensky (Prensky, 2001) identified eight types of digital games:

1. Action games
    a. Arcade games (Pinball, Pac-man)
    b. Side-scrolling games (Defender)
    c. Platform jumping games (Super Mario Brothers)
    d. Maze games
    e. Third-person shooters (Quake)
2. Adventure games (Zelda, Zork)
3. Fighting games (Mortal Kombat)
4. Puzzle games (Tetris)
5. Role-playing games (Lara Croft: Tomb Raider)
6. Simulation games (Flight Simulator, The Sims)
7. Sports games (FIFA football)
8. Strategy games (Lemonade Tycoon, Team Hospital)

2.4 BRIEFLY C++

The C++ Programming Language is a programming language that is extended from the C Programming Language. The C Programming language was designed and implemented in the year 1972 at Bell labs by Dennis Ritchie, at the same time the UNIX operating system was being developed there. C was a direct descendant of the language B, which was created by Ken Thompson as a systems programming language for the fledgling UNIX operating system. B, in turn, was based on a simple systems language, BCPL which was designed and implemented in the 1967 by Martin Richards while at MIT.

It is unfortunate that neither BCPL nor B is a typed language. Not typed language mentioned here means that all data are considered as machine words, which leads to many complications and insecurities. These problems led Dennis Ritchie at Bell Labs to extend the B language into a new type of language which was originally known as NB, for "New B". Ritchie credits some of his changes to language constructs found in Algol68, especially obvious in the “for” and “switch” statements, assigning operators, and treatment of pointers. Prior to that, Ritchie gave his new language a name: "C".

In a memorable year of 1980, the first step from C toward C++ was made by Bjarne Stroustrup at Bell Labs. C++ was designed for the UNIX system environment. It represents an enhancement of the C programming language. Modifications can be seen in the function parameter type checking and conversion, and not to forget, classes.

From the day it was created, C++ continues to grow and develop into a very popular language. Even till today, it is still widely being used. Factor that promotes to its popularity include the availability of good and inexpensive compilers, compatible with C and it too enables programmers to improve the quality of code produced, thus making reusable code easier to write.

2.5 SOFTWARE AND TECHNOLOGIES

2.5.1 System Platform

2.5.1.1 Windows 98

Windows 98 is an improved version of Windows 95. It has 23 bit and 16 bit code nativity that is equally the same to Windows 95. The features that Windows 98 offers are:

- Has embedded MS-DOS within it
- User friendly graphical user interface
- Integrates Internet Explorer into its graphical user interface
- Satisfactory array of driver support for hardware
- Better memory management compared to Windows 3.x and Windows 98
- Easy to use and install
- Support multimedia application and DirectX
- Support Universal Serial Bus function
- Provides more stability, better performance and better system tools

2.5.1.2 Windows NT/2000/XP

Windows of these versions are more stable and provides better security than the previous versions. They support full 23 nativity code and remove the need of MS-DOS. Though slower in booting up, it is compensated by providing good network connectivity. These Windows versions have improvised multimedia support, better interfaces with icons and more stability as less system crash is seen. Other than that, they have higher connectivity to plug and play hardware especial Universal Serial Bus (USB) devices. Features include:

- No more DOS. Instead has very own virtual DOS environment called Network Transmission (NT) Virtual Dos Machine which will run DOS application inside it.
- More user-friendly and easier to use – provides icons and better graphical user interface layout
- Network operating system
- Improved security
- Have license fees whereby people who would like to use these versions of Windows have to pay in order to use.

2.5.1.3 UNIX

This operating system is originally developed by AT&T for mini computers. It can be used on many computer system types and platforms for both personal computers and mainframes. There are many variants of UNIX. This includes UNIX System V from UNIX Systems Lab, Solaris from Sun Microsystem and AIX from IBM.

2.5.1.4 Linux

Linux is an almost similar UNIX operating system. It was created by Linus Torvalds at University of Helsinki with the assistance of developers throughout the world. It is developed under the GNU, General Public License and its source code is available freely to everyone. Red Hat Linux 6.1 is a network operating system is one of the many variants of Linux operating system.

2.5.1.5 MAC OS

Mac has two types of operating system, (1) Mac OS 9 and (2) Mac OS X ("Ten"). Basically, these two operating systems are a proprietary Apple operating system. Apple computers are quite popular especially in the field of publishing, education, and graphic arts. Many people argue that Mac OS from Apple Computers have always been the easiest and most intuitive of all operating systems.

2.5.1.6 Comparisons of System Platforms

Table 2.3 shows the comparison of Linux Red hat, UNIX, Microsoft Windows and Macintosh (abbreviation, MAC) system platforms.

Table 2.3: Comparison of System Platforms

	Linux Red Hat	UNIX	Microsoft Windows	Macintosh
Overview	• Robust and quick • Full customizatio n for web functions	• Designed by programmers for programmers • Access and share time and resources	• Popular • Capable of supporting web application	• Support many application
Installation issue	• Need concept on disk	• Need concept on disk	• Easy to install using friendly	• Easy to install using

	partitioning and mounting file system	partitioning and mounting file system	interface wizard	friendly interface wizard
User friendly	• Not friendly. Too cryptic	• Not friendly. Too cryptic	• Friendly with windows based interfaces	• Friendly with Mac-based interfaces
Security	• Vulnerability high • List of codes available	• Vulnerability high • List of codes available	• Vulnerability Low. Application not truly available in the Internet	• Vulnerability Low. Application not truly available in the Internet
Cost Effective	• Freeware	• Not cost effective. With modification, not the whole OS needs to be recompiled	• Cost effective. Depends on the version	• Cost effective. Budget around $129 for a user
Scalability	• Multitasking	• Multitasking	• Symmetric multiprocessing	• Multitasking
Stability	• Stable	• Stable	• Sometimes not.	• Stable
Compatibility with web development tools	• Low. Does not offer much tools	• Incompatible	• Compatible	• Compatible

2.5.2 Programming Languages and Technologies

2.5.2.1 Markup Language

2.5.2.1.1 HTML

Hypertext Markup Language, more familiarly known as HTML is a language used to specify the construction of a web page. It is stored as standard ASCII (American Standard Code for Information Interchange) files. It is an efficient and easy-to-use language. It enables a programmer or web site builder to request a web browser to

format and display web pages in a particular way. There are 2 most useful elements of HTML (www.bellcraft.com/mash/main.htm):

1. Describe the format of website
2. Define information about the document

2.5.2.2 Web Application Programming Languages

2.5.2.2.1 Active Server Page (ASP)

Active Server Page or ASP in short is a technology from the Microsoft Corporation that provides the capability for web server to process application logic and then delivers standard HTML to the client browser. It is generally faster in execution and thus, saves time when retrieving information. It is feature rich and has high stability compared to other web programming languages. Its advantages include:

- Enable user to dynamically edit, change or add any content of the webpage
- Response to user's queries
- Access any database and return result in the browser
- Enable customization of webpages
- ASP can be viewed in any browser because ASP files returned as plain HTML

2.5.2.2.2 ASP.NET

ASP.NET is a programming framework built on the common language runtime that is mostly used on a server to build great Web applications. It is feature rich and has flexible architecture, designed to cater the needs of current development and also future development. It offers many advantages over existing Web development frameworks:

- Enhanced performance because ASP.NET is a compiled common language runtime code running on the server
- Provides world class tool support
- Gives power and flexibility to programmers
- Its simplicity in performing common tasks
- Manageability because it adopts text-based, hierarchical configuration system
- Provides scalability and availability, together with customizability and extensibility
- Most important of all, security

2.5.2.3 Web Scripting Language

2.5.2.3.1 JavaScript

JavaScript is a general purpose language that is designed to let programmers of all skill levels control the behaviour of a software object. It was developed by Brendan Eich of Netscape. You may get it confuse with another language that is so similar in name; Java programming language. Both languages can be considered as powerful language and furthermore, both have almost similar syntax. However, these 2 languages were developed independent of each other and for different audiences. JavaScript is mostly used in web browsers where software objects represent a vast range of Hypertext Markup Language (HTML) documents and elements of documents.

2.5.2.4 Multimedia Programming Languages

2.5.2.4.1 ActionScript

ActionScript is the programming language built especially for Macromedia's Flash MX application. It enables you to provide more than just linear animation instructions with the main timeline. With ActionScript, Flash movies are never boring because it allows reaction to events as a result of user interaction with the movie itself. ActionScript takes control of what the users sees by providing more than just linear presentations.

2.5.2.4.2 Lingo

Lingo is a powerful programming language built inside Macromedia's Director application. It is a set of rules to manage media.

2.5.3 Web Application Development Tools

2.5.3.1 Image Creation & Editing Tools

2.5.3.1.1 Adobe Photoshop Elements

Adobe Photoshop certainly is not a graphic editing tool that is alien to us. It is one of the leading image manipulation tools available in the world-wide market today. It is used especially to create complex effect on scanned or downloaded images or graphics. It offers layering and filtering techniques that enable user to create unimaginable and cool images. Besides that, it offers efficient image editing through retouching, painting, blurring and etc. tools.

2.5.3.2 Digital Sound Editor, Application & Tools

2.5.3.2.1 Audacity

Audacity is a free program especially for sound editing. It is written by a group of volunteers around the world who are dedicated to sound editing applications and tools. It is almost as good as infamous audio editing software such as cubix and sonar. It allows you to import audio track and then modify tracks by adding effects to the track itself.

2.5.3.3 Animation Tools

2.5.3.3.1 Macromedia Flash MX

It is almost similar to Macromedia Director in terms of metaphors and components. The main components in Flash MX are vector graphics and sounds. It incorporates interactivity to the point where it allows viewers to provide input. Other than that, it allows the creation of non-linear movies that can interact with other applications.

Flash movies are small and compact. One of its advantages associated with the size of the movies created is that it can be downloaded rapidly. Besides that, it can be scaled or be scaled to viewer's screen size.

Flash is everywhere. What is better is that it is designed solely for the Web promising good deals for people who want to delve more into the world of Web.

2.5.3.4 Authoring Tools

Authoring means assembly and synchronization of all media components or elements that have been prepared for the multimedia application. Authoring tool is an utility or application software used to compile all these elements.

Authoring tools can be categorized into three type namely Page-based tools, Icon-based tools and Time-based tools.

2.5.3.4.1 Page-based tools

The working metaphor of page-based tools is based on a book or page. Its objective is to create a book-like interactive multimedia system. So, elements are organized into pages similar to a book or card file. Pages or card files can be linked to other pages or card files in an organized sequence.

2.5.3.4.2 Icon-based tools

The working metaphor of icon-based tools is flow charts. Its objective is to create a map of how various multimedia elements are connected together and their characteristics of each individual element. An example of icon-based tools is Macromedia Authorware.

2.5.3.4.2.1 Macromedia Authorware

Macromedia is a name that is really well-known in the field of multimedia. Macromedia Authorware is one of Macromedia's vast ranges of software applications that provide authoring function for creating web and online learning.

2.5.3.4.3 Time-based tools

Time-based tools rely on the usage of cells, frames and channels. Multimedia elements are sequentially organized into frames that can be played-back to the user. Since all the elements are related to frames, it is also sometimes known as 'Frame-based tools'. An example of time-based or frame-based tool is Macromedia Director and Macromedia Flash.

2.5.3.4.3.1 Macromedia Director MX

Macromedia Director like Macromedia Authorware is also an authoring tool. It comes with a full-blown programming language called "Lingo". It adopts theater and movie metaphor and employs concepts such as Cast, Score and Stage. It is especially suitable to be used to create animations. Resulting output of Macromedia is known as 'movies'. It can be compressed, embedded and displayed in a Web page using a plug-in known as 'Shockwave'.

2.5.3.5 Web Editors

2.5.3.5.1 Macromedia Dreamweaver MX

Macromedia Dreamweaver is a feature-rich program for professional Web design and development. Macromedia Dreamweaver MX provides:

- Integrated environment that allows user to design, build, and manage Web sites with friendly visual design tools
- Customizable coding environment that support technologies for both static Web sites and server-based Web applications.
- Web sites can be built in the form of ASP pages. Simple ASP pages can be built without the fore knowledge of Visual Basic or even SQL.
- It can be integrated with multiple knowledge technologies and supplies visual tools to greatly enhance a developer's productivity.
- Support "What You See Is What You Get" (WYSIWYG)
- May create extremely complex web pages
- Has tools that are intuitive and easy to use

2.5.3.5.2 Microsoft Frontpage

It is one of the many products of Microsoft Corporation. It is an application for editing and publishing webpages. Microsoft Frontpage provides:

- User-friendly environment
- Easy to use
- Supports HTML
- Has Object Database Connectivity (ODBC) that provides connection to database

- Support "What You See Is What You Get" (WYSIWYG) that allow images onscreen to appear exactly the same in print out.
- Has integration with Microsoft Office and Microsoft Visual Basic
- Supports multimedia components and animation

2.5.4 Web Server

Web server is a kind of network server with special characteristics for World Wide Web (WWW) networking. It handles permission who can do what in the web page, execute program, communicate with and keep track of directories and maintain communication with client computers which can make requests for files and actions from the server's computer.

2.5.4.1Personal Web Server (PWS)

PWS is a desktop server that enables developer to publish their pages, share document with workgroup and/or build web application. Advantages of using PWS:

- Provides good test drives for Internet Information System (IIS) website
    - Supports ASP, script debugging
    - Great platform for testing before hosting the site
- Wizards
    - Helps in developing through setting up homepage and sharing files

2.5.4.1 Internet Information Server (IIS)

Web services needs to be interwoven to the mainstream of business computing. Windows 2000 Server operating system included an updated version of IIS. IIS 5.0, the newer vesion of IIS, provides:

- Reliability and performance
- Management whereby it is easier to install and maintain
- Security with security protocols such as Digital Authentication, Server Gated Cryptography, Kerberos V5 Authentication protocol and Transport Layer Security.
- Application services

2.5.5 Web Browser

2.5.5.1 Internet Explorer 6.0

Internet Explorer is a set of core technologies in Microsoft Windows XP Home Edition and XP Professional. It has new and advanced feature that is interesting and efficient. It maintains the privacy of your personal information on the web. Furthermore, Internet Explorer Service Pack 1 has the latest update on the Internet Explorer. Features included are web privacy, flexibility and reliability.

2.5.5.2 Netscape Navigator

2.5.5.3 Comparison between Internet Explorer and Netscape Navigator

Table 2.4 shows the comparison between Microsoft Internet Explorer web browser and Netscape Navigator web browser

Table 2.4: Comparison between Internet Explorer and Netscape Navigator

	Netscape Navigator	Internet Explorer
System Platform	• Supported all Windows Operating System, MAC and UNIX	• Only supports Microsoft Windows Operating System and certain MAC and UNIX
Speed	• 35% faster in caching Web pages • Even faster when cache is empty	• Faster at loading large files • More efficient in managing large image that were already loaded into the memory
Multimedia	• Support vast range of media • Requires separate plug-in for each format • However, it comes with preconfigured and coupled with more popular plug-ins	• Has equal support • Has ActiveX feature called Active movie that handles all .AU, .AVI and .AIFF files. • Support streaming
Security	• Has same level of security for online transaction • Provide support for 128 bit keys for security code • Can warn users when a Web server attempts to leave cookies on their system	• Has all the features in Netscape Navigotor • Provides more security with Authenticode which protects user from malicious downloads of ActiveX controls or Java applets. • Support parental controls and rating system (PICS and RSAC)
Java	• Has Just-in-Time Java Compilers	• Has Just-in-Time Java Compilers • 15% faster in performing computations • Runs Java faster on pages where Just-in-

		Time compiler provides almost no benefits
HTML	• Has own proprietary extensions to HTML specifications	• Has own proprietary extensions to HTML specifications

2.5.6 Database System

2.5.6.1 Web Database Management System

Web database generally refers to data stores or information repository that allows access through a query language or application programming interface (API). In most web-based application, database serves as the basic building blocks for information servers. They are use for multiple reasons. Reasons are:

- Better management of large, document-based information repositories to internal and external users of the information
- Unlock the potential of unused information in the organization database. Database does not have to be physically located with the users of the database application

Extend the function of web servers, so that information maintained will be made available to the public.

2.5.6.1.1 MySQL

Structured Query Language, in short SQL is a standard language for relational database management system. In other words, it is a language used to communicate with databases. SQL statements allows tasks to be perform on the database such as update,

data in the database, retrieve data from the database. Common databases that uses SQL includes Microsoft SQL Server, Access and etc.

MySQL database server is the most popular open source database throughout the world. It has become the core of many high-volume, business-critical applications because it saves time and at the same time cost-friendly.

2.5.6.1.2 Microsoft SQL Server

It is a significant tool from data warehousing to application that requires not only large amount of information but different simultaneous users as well. On the whole, Microsoft SQL Server 2000 is a great and powerful database management system that provides compact databases for rapidly developing application that extends enterprise database management capabilities. It is able to manipulate data from the client side, validate data before it enters the database at the server side. It also allows publishing database information in HTML documents, and at the same time build active web site.

2.5.6.1.3 Microsoft Access

Microsoft Access is part of many applications that is produced by Microsoft. It provides a set of tools for both professional developers and novice developers. It enables powerful database solutions to be created that make organizing, accessing and information sharing efficient and easier than ever. Basically, it is a relational database management system that is used to create relational databases. The advantages of Microsoft Access:

- Support a wide range of data formats
- Links business systems easily
- Incorporate SQL Server data on your Access solutions so that you can get most out of your enterprise data
- Reduce the number of errors and provide help when needed
- Enables you to share information efficiently
- Facilities to import, export and link tables to HTML files to process
- Forms and datasheets can be saved as ASP

Fully networkable, widely-used and tested

2.6 CURRENTLY AVAILABLE C++ LEARNING SYSTEM

It is important that I highlight that there are no current game-based C++ learning system in the market today. Learning system available are mostly e-books in the form of CD-ROMs and online tutorials and notes.

Below are websites that offers C++ tutorial and notes.

1. http://www.cplusplus.com/

   The content of this site is written by Juan Soulie for C++ Resource Network. It offers a user manual under the heading "Introduction" where it teaches users how to use this site for C++ programming. The author divides the whole C++ syllabus into 6 main chapters, excluding epilogue. Tutorials provided in this site are in the form of source codes which can be copied and compiled with almost any C++ compiler.

2. http://www.intap.net/~drw/cpp/index.htm

Basically, this site is almost similar to the site above. It arranges the C++ syllabus in a table of content that makes it easier for the user to navigate around the chapters he/she would like to learn. This site provides tutorials in the form of slides shows. Other than tutorials, it also provides a search function for the user. A glossary of terms is also made available to the user.

2.7 EXISTING GAME-BASED LEARNING SYSTEM

2.7.1 Monkey Wrench Conspiracy

Monkey Wrench Conspiracy (Monkey Wrench Conspiracy, 1999) video-game based tutorial or a simulation for a complex technical product, designed to teach industrial engineers how to use new 3-D design software. It is recorded to have distributed over 1 million copies world wide.

In this tutorial, the player act as an intergalactic secret agent dispatched to deep space to rescue the Copernicus Station from alien hijackers. In order to succeed in the game, the player must use the think3 software provided to design everything needed for the mission.

This videogame tutorial provides simulation to that enables learners to "feel" and enable them to cope with almost similar real-life tasks and projects.

Figure 2.2 shows some of the screen shots of Monkey Wrench Conspiracy software.

Figure 2.2: Screenshots of Monkey Wrench Conspiracy Video Game Tutorial

The main advantage of this software is that it puts the player into the situation that is simulated by the software which indirectly guides the player to use new 3-D design. It is highly interactive and encourages critical thinking.

This videogame employs "Discovery Learning" approach whereby the player learns with a combination of questions, tasks, backup manual and videos as needed. An interesting feature of this videogame tutorial is that it includes a non-game alternative for those who do not like games or do not want to play games at that particular time.

2.7.2 I Love Math!

"I Love Math!", from the CD-ROM title itself, is a software especially to learn and enhance a child's level of math comprehension. It is an amazing animated time-travel adventure that will keep a child, especially in grade 2 through 5, interested to learn mathematics. It provides a blend of solid, curriculum-oriented content and incentive-driven game play. Syllabus provided by this software application includes:

- Measurements
    - Learn through a series of puzzles and mysteries
- Fractions
    - Learn by laying pipes to save an underwater kingdom

This software application received an award from the Children's Software Revue Education Software EDDIE Award and also approved by the National Parenting Center

Seal of Approval. Furthermore, it is recommended by consumerguide.com and thereview.zone for parents around the world, who wants their children to master math.

Screenshots of "I Love Math!" are shown below (Figure 2.3):

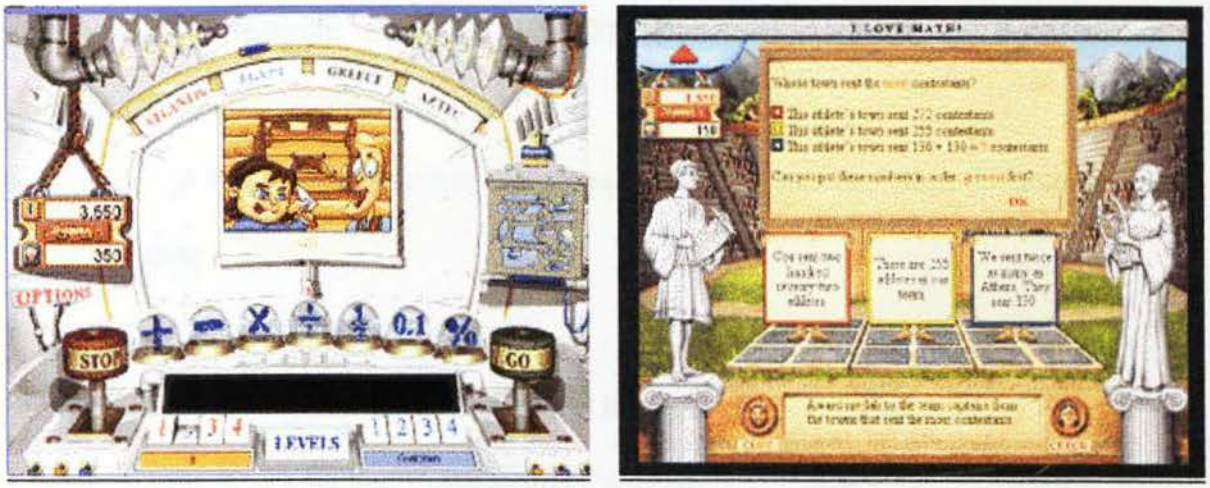


Figure 2.3: Screenshots of "I Love Math!" software

2.7.3 Jumpstart Advanced 1st Grade

Jumpstart Advanced 1st Grade is a 3-CD set especially to prepare your child for 1st grade and beyond. It provides premium features and extensive content that keeps kids engaged and learn for hours and hours.

Jumpstart Advanced 1st Grade enables kids to explore from simple spelling to science. Special features that Jumpstart Advanced 1st Grade have includes:

- Auto-leveling that allows kids to learn at their own pace

- Introduce different learning styles that cater for a variety of kids. Uses all 7 identified learning styles
- Reading – Phonics, Consonant sounds, Blends, Digraphs Short and Long Vowels, Syllables, Vocabulary, Reading Comprehension
- Writing – Sentence Building, Spelling, Punctuation, Parts of Speech and Alphabetizing
- Math – Addition, Subtraction, Counting Coins, Fractions, Skip Counting and Sequencing
- Science – Life, Earth and Nutrition
- Social Studies – Communities and World Regions
- Art – Colours, Shapes and Patterns, and Creativity
- Music – Melody, Rhythm, and Instrument Families
- Thinking Skills – Problem Solving, Sorting by Multiple Attributes and Patterns

Screenshots of Jumpstart Advanced 1st Grade (Figure 2.4):

Figure 2.4: Screenshots of Jumpstart Advanced 1st Grade

2.7.4 ABC Kid Genius Ver: 2.0

ABC Kid Genius is CD-ROM based software that teaches children basic alphabets, numbers, counting, reading and spelling using sound, pictures and video clips. Young children and toddlers learn by watching lesson. Furthermore, they are taught to use and then, progress to type alphabets using the keyboards.

ABC Kid Genius has story reading functions, flash words and spelling functions. Parents can even create own Spelling words using own pictures, voice or even video clips. A demo version of ABC Kid Genius is downloadable from the Internet.

Figure 2.5 shows the main interface of ABC Kid Genius software:

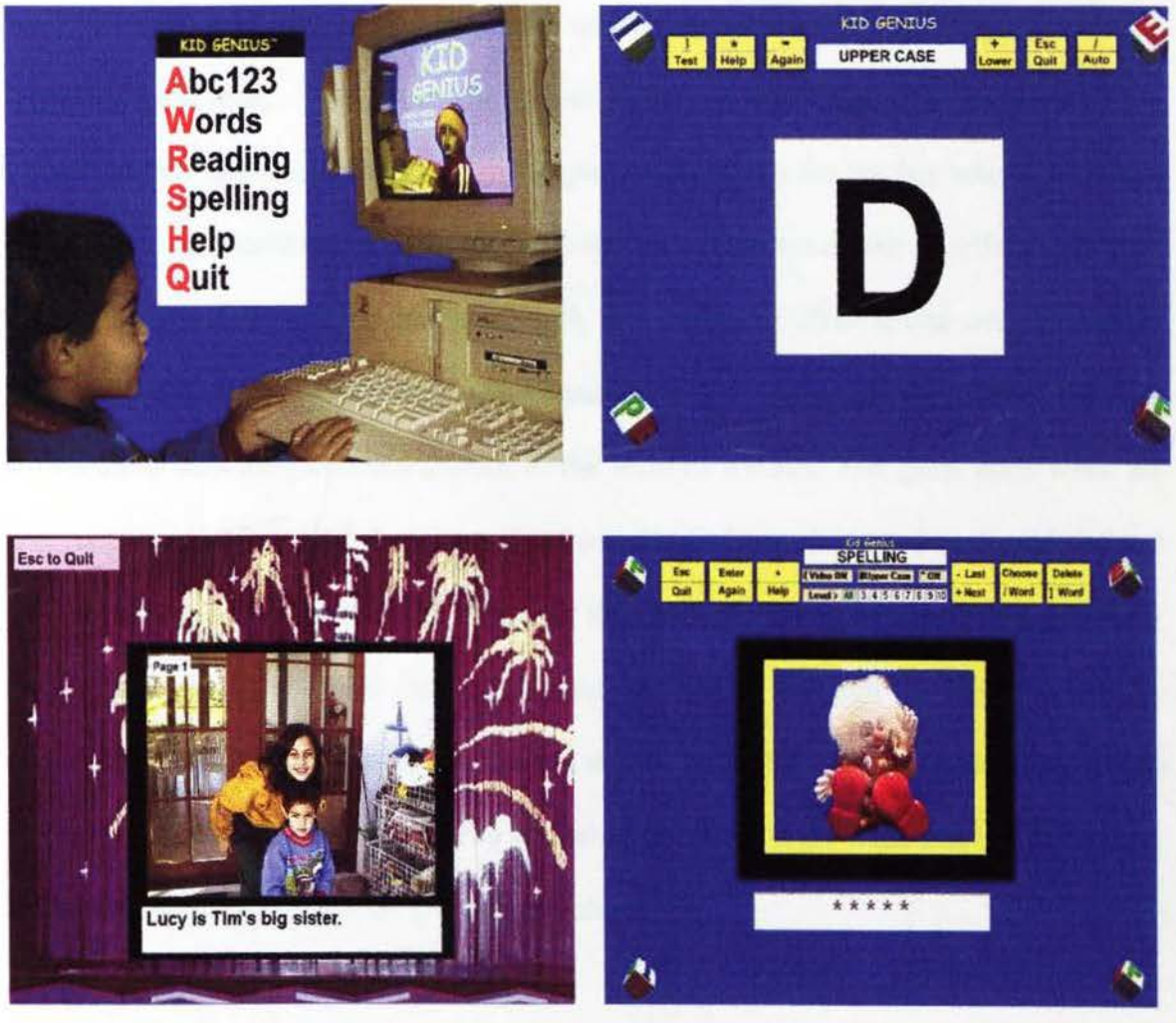


Figure 2.5: Screenshots of ABC Kid Genius

2.7.5 UniGame: Social Skills and Knowledge Training

"UniGame: Social Skills and Knowledge Training" (UniGame, 2003) is a framework that provides the possibility for interested teacher to apply game-based learning to his/her classes. Its aim is to be used additionally to regular traditional or online classes.

It is a web-based role playing game, not the usual digital game we are visualizing. It is based on constructivist learning approach and collaborative learning. It is a game that supports multi-users. The players form four teams, in which each team consists of 6 players at the maximum depending on the game topic set by the teacher who at the same time, act as the moderator. Players are supposed to comprehend their specific role inside their team and have an argumentation with the players of other teams over a specific subject (subjected to the theme). The players gain knowledge on the subject through information searching and discussions in the form of forums. The game ends when all the specified sub-parts of the selected subjects have been discussed and goals of the players are reached. The amount of points they can win from the discussion has been specified before the game started. Points are given through a "chip allocation procedure". Each team allocates 100 points in total to 3 of the six available sub-parts of the subject that will be discussed. The winner of the game is the team that has gained the highest number of points upon game completion.

An example of game usage is a game with theme "Tunnel Building". The teacher wants his/her students to reflect actively upon interdisciplinary consequences and ethical behaviour of engineers. Aim of the game is that the four teams compete to make the best offer and technical solution to build tunnel on the defined location. During the game

itself, the team members are allowed to search for relevant information. Teams are also expected to react to unexpected new conditions. During general discussion, different important subjects are discussed and a consensus on which a solution is the most appropriate should be achieved. A screenshot from the game is provided in Figure 2.

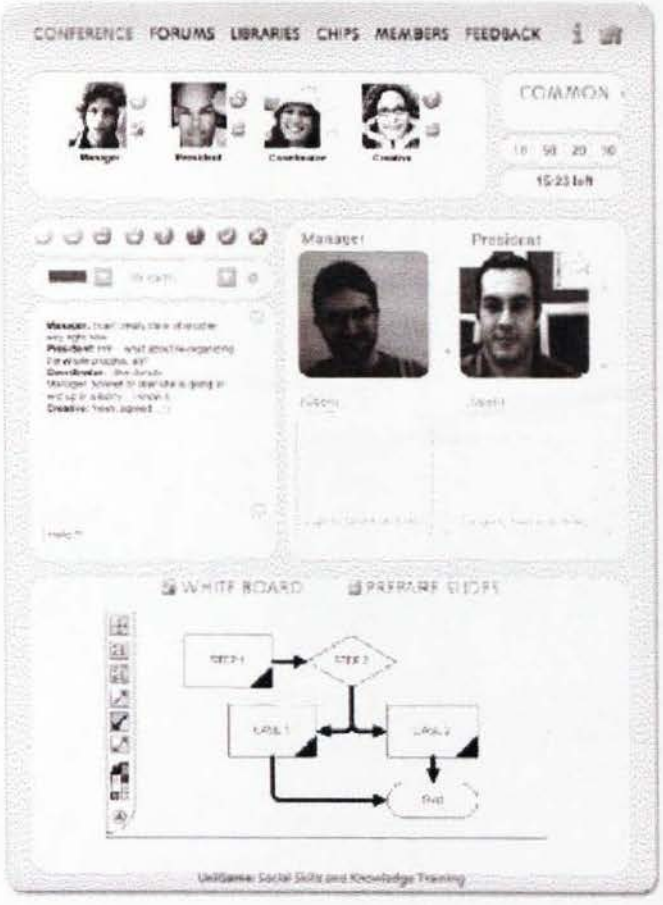


Figure 2.6: Screenshot from the Virtual Conference of UniGame

2.8 SUMMARY

This chapter basically looks at multimedia and games in the context of education - web-based learning. Other than that, a detailed study were carried out on software and technologies available and related to the building of the digital game-based C++ learning system. Researches on existing system proved to be useful to provide insights, especially on the games part.

CHAPTER 3:

METHODOLOGY

CHAPTER 3: METHODOLODY

3.1 INTRODUCTION

People including computer-techies have problems using and telling apart the terms "methodology" and "life-cycle". These two words are used interchangeably and at most of the time, incorrectly. System development processes are derived from a natural system life-cycle, which means a life-cycle just happens (Hodges & Sasnett, 1993). Systems development methodology, in contrast, "executes" the systems development stage of the system life-cycle (Hodges & Sasnett, 1993).

To further clarify the term "methodology", system development methodology can be defined as a formal and standardized system development process that defines a set of activities, methods, best practices, deliverables and automated tools for system developers and project managers are to use to develop and continuous maintain most or all information system and software (Stair & Reynolds, 2001).

In the following sub-chapters, several life-cycles models will be introduced and describe. This includes Waterfall Life-cycle Model, Rapid Prototyping, Spiral Life-cycle Model, Interactive Multimedia System Design and Development (IMSDD) Model and Unified Process.

3.2 LIFE-CYCLE MODELS

In an ideal world, development of software product can be depicted as a structured process shown in Figure 3.1 (this figure is adapted from Schach's book entitled "Object-Oriented & Classical Software Engineering" (Schach, 2005)).

The system is developed from scratch (starts with nothing) by firstly, determining the client's requirements of the system and followed by analysis on the requirements gathered. Next, based on the analysis, a design of the system is produced. Subsequently, the completed software product is implemented.

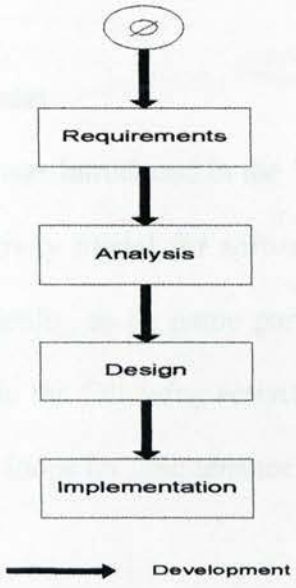


Figure 3.1: Ideal Software Development Process

It is important to take note that in the real-world, software development is different. According to Schach (Schach, 2005), there are 2 reasons why practical software

development and theoretical software development is poles apart. First and foremost, software professionals are human and human with their human behaviour, often make mistakes. Secondly, client's requirements on the system can change inconsistently while the software is being developed.

The following sub-topic will give a brief explanation on some existing life-cycle model available today. Life-cycle model that will be discussed are (1) Waterfall Life-cycle Model; (2) Rapid Prototyping Life-cycle Model; (3) Spiral Life-cycle Model; (4) Interactive Multimedia System Design and Development Model; and (5) Unified Process.

3.2.1 Waterfall Life-Cycle Model

The waterfall life-cycle model was introduced in the 1970s by Royce (Royce, 1989). It is developed as an overall activity model for software engineering. There are many variants of this model but basically, as its name portrayed, outputs from one process activity in the model *cascade* to the following activity. Figure 3.2 shows the waterfall life-cycle model with feedback loops for maintenance.

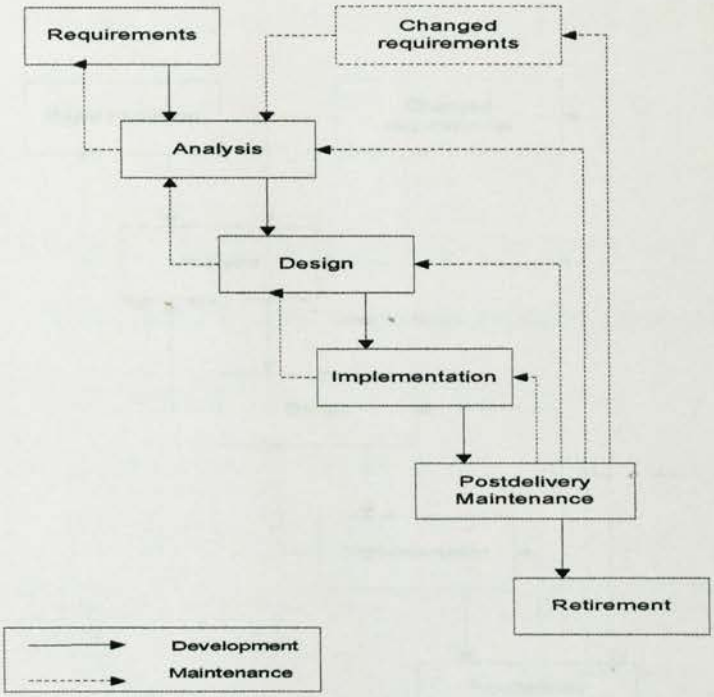


Figure 3.2: Waterfall Life-cycle Model with feedback loops

3.2.2 Rapid Prototyping Life-Cycle Model

Prototype is a small scale, representative, or an incomplete working model for a desired software product (Stair & Reynolds, 2001). As its name suggests, rapid prototyping is a technique which supports quick development of a functioning but incomplete model of a software product using rapid application development tools. It is an iterative approach to system development process in which speed is an essential criterion of the development.

Its life-cycle model is almost similar to the Waterfall life-cycle model with maintenance capabilities. Figure 3.3 shows the life-cycle model for rapid prototyping

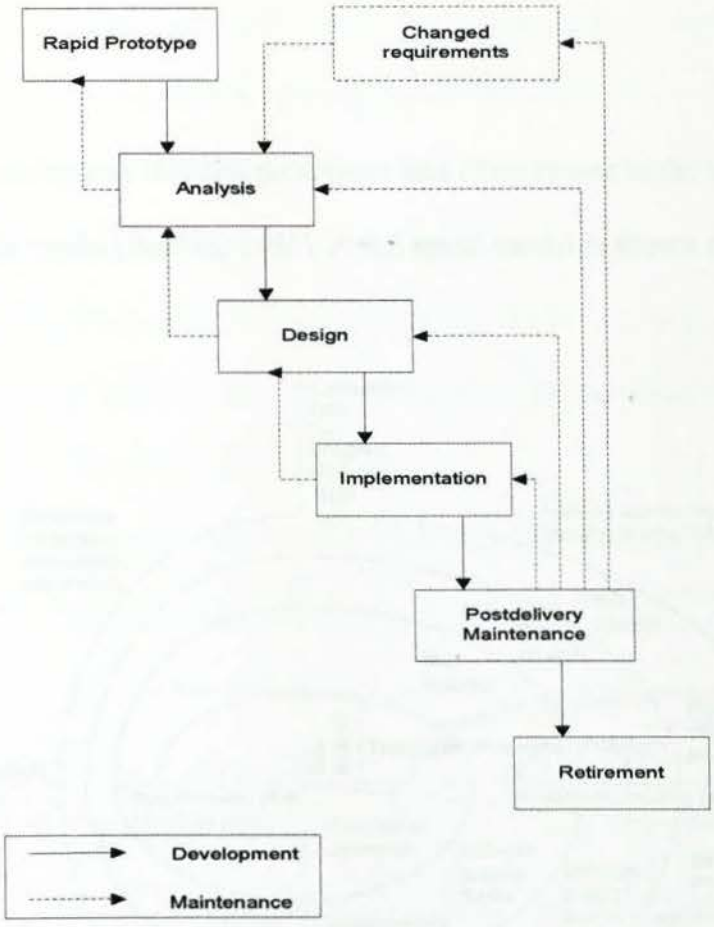


Figure 3.3: Rapid Prototyping Life-cycle Model

3.2.3 Spiral Life-Cycle Model

Risk is always a part of any software development. It is the hope of every software developer to minimize risks whenever it is possible. One of the ways to minimize the probability of risk is to construct a prototype, named proof-of-content prototype. This prototype is not of an equal to the prototype built in rapid prototype life-cycle. Instead,

it is more of an engineering prototype where it functions only to test the feasibility of construction.

The idea of minimizing risk through prototypes and other means is the idea underlying the spiral life-cycle model (Boehm, 1988). A full spiral model is shown in Figure 3.4.

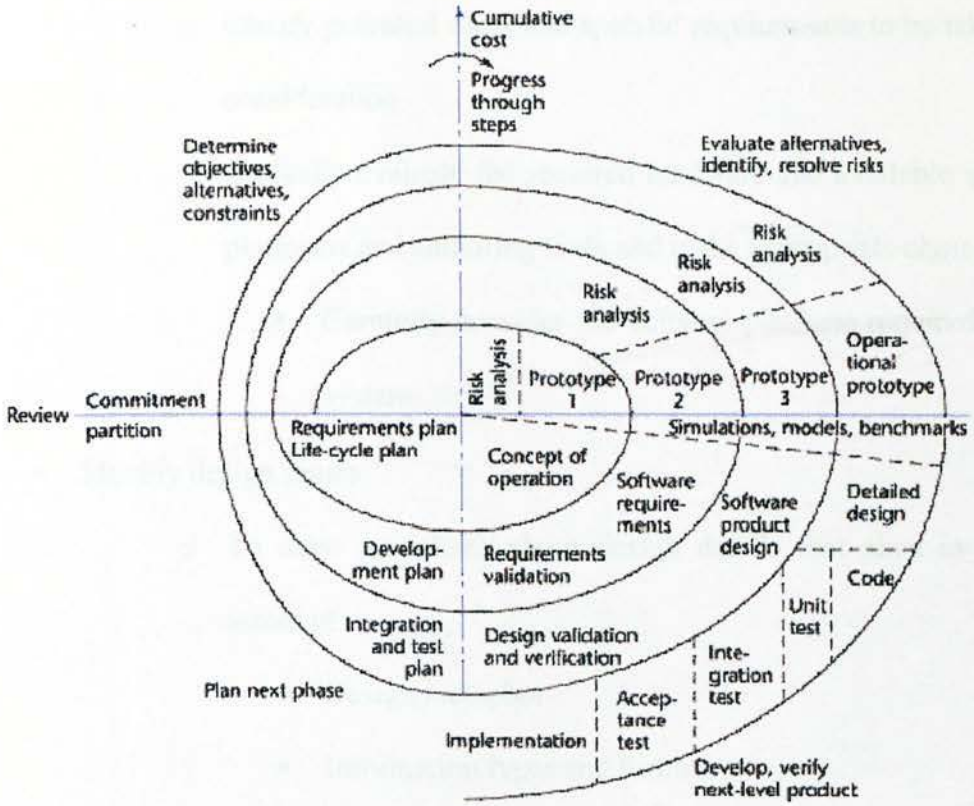


Figure 3.4 Full Spiral Life-Cycle Model

3.2.4 Interactive Multimedia System Design and Development (IMSDD)

IMSDD is a detailed approach towards media integration. It is especially convenient and suitable for handling and synchronizing media. Its life-cycle model has 4 main stages or phases.

- Identify requirements
    - Provides system definition which includes aims and objectives
    - Clarify potential users and specific requirements to be taken into consideration
    - Critically evaluate the required hardware and available software platforms and authoring tools and make appropriate choices
        - Carefully consider the delivery platform required for the system
- Identify design issues
    - To draw guidelines about design details that aims to address issues of
        - Design metaphor
        - Information types and formats
        - Navigational structures
        - System controls
- Implementation
    - Once design features have been defined, implementation can begin.

- Can be done using authoring tools
- Consists of
  - Creating prototype of the system
  - Beta-test the prototype for possible design and control problems

- Evaluation
  - Tested and evaluated against stated goals and objectives

Figure 3.5 shows the IMSDD life-cycle.

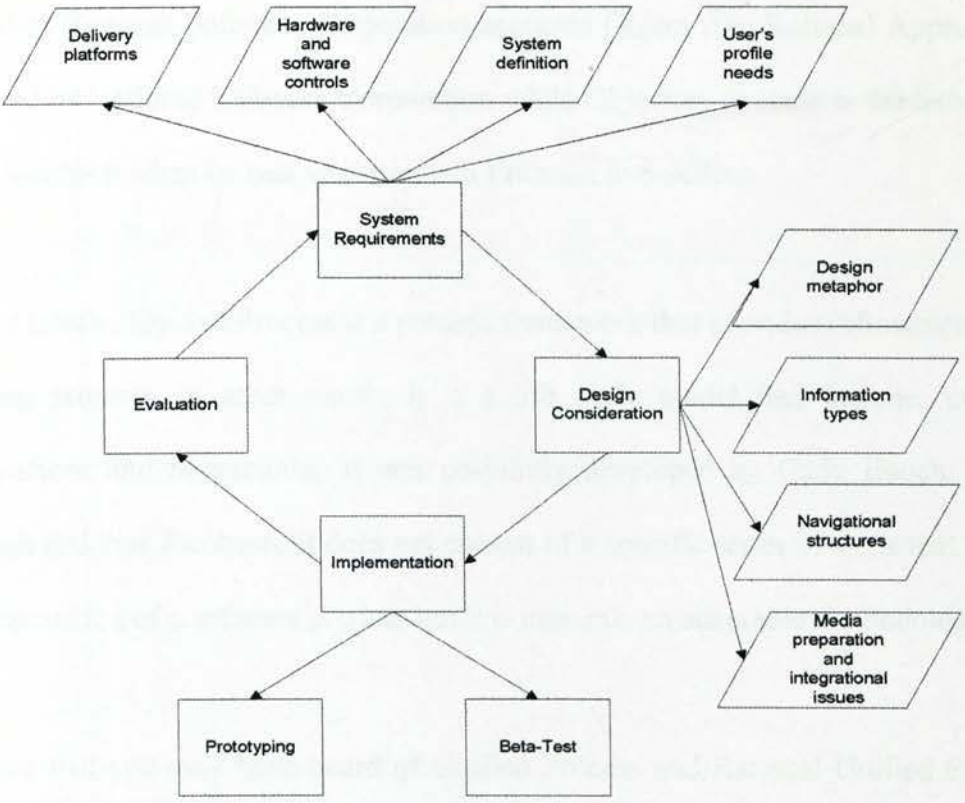


Figure 3.5: IMSDD Life-cycle Model

3.2.5 Unified Process

Unified Process is the primary object-oriented methodology used in most software production today (Schach, 2005). It is use-case driven, architecture-centric, iterative and incremental development process (Sinan Si Alhir, 2002). It is applicable not only to large-scale project but also small-scale as well in which projects may range from managerial to technical.

Unified Process, or in short UP, emerged as a result of mergence between Rational Approach, by Rational Software Corporation and Objectory, by Object AB in the year 1995 when Rational Software Corporation acquired Object AB. Rational Approach is developed by Rational Software Corporation while Objectory Process is the handmade of Ivar Jacobson when he was working with Ericsson in Sweden.

To put it briefly, Unified Process is a process framework that provides infrastructure for executing projects. In other words, it is a life cycle model that includes context, collaborations and interactions. It was co-jointly developed by Gady Booch, James Rumbagh and Ivar Jacobson. It does not consist of a specific series of steps that results in a construction of a software product but it is certainly an adaptable methodology.

I am sure that you may have heard of Unified Process and Rational Unified Process. Most people get confused between these two. So, basically, Rational Unified Process is a process product developed by Rational Software Corporation that provides details for

executing projects using UP methodology. It is commercial product that provides guidelines, templates and tools for the UP framework.

Unified Process has several core phases:

- Inception phase
    - Determine the scope for the project
    - Determine the problem to be solved by the project and its objectives
    - Identify people that interact with the system
    - Identification of risks also must be carried out to mitigate them.
- Elaboration phase
    - Establish requirements and architecture for the software development project
    - Major activities are refinements or elaborations of the previous phase.
- Construction phase
    - Growth of the system
    - Production of the first operational-quality version of the software product.
    - Focuses on implementation and testing of the system
- Transition phase
    - Supply the system to the end user
    - Ensure that all requirements have been met.

Within each phase are a number of iterations. Iteration within the phases is a complete development life-cycle – waterfall model. So, in other words, there are 5 workflows that cut across the set of four phases. These workflows are identified as Requirements workflow, Analysis workflow, Design workflow, Implementation workflow and Test workflow. So, every step performed falls into one of five core workflows and transition phase. Figure 3.6 shows the Enterprise Unified Process life-cycle model.

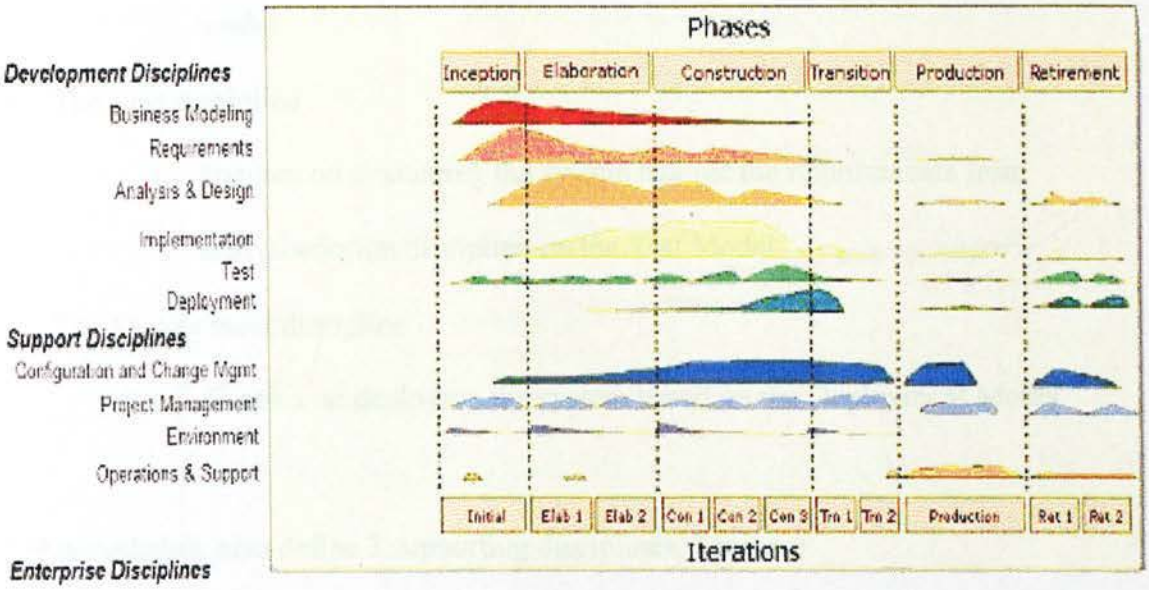


Figure 3.6: Enterprise Unified Process Life-Cycle Model

UP also defines the following 6 major disciplines:

- The Business Modeling discipline
    - Focuses on understanding the business automated by the system
    - Captures knowledge in a Business model
- The Requirements discipline

- Focuses on understanding the requirements of the system
- Captures such knowledge in Use-case model

- The Analysis Design discipline
  - Focuses on analyzing requirements collected and designing the system
  - Captures such knowledge in an Analysis/Design Model
- The Implementation discipline
  - Focuses on implementing the system based on the Implementation model
- The Test discipline
  - Focuses on evaluating the system against the requirements from analysis/design discipline on the Test Model
- The Deployment discipline
  - Focuses on deploying the system based on the Deployment Model

UP nonetheless also define 3 supporting disciplines. They are:

- The Configuration Change Management discipline that focuses on managing the configuration of the system and change requests
- The Project Management discipline that focuses on managing the project as a whole
- The Environment discipline that focuses on the environment of the project

From the Figure 3.6, you can see the distribution of efforts across phases, iterations and disciplines that focused on addressing business and technical risks. During Inception

phase, most effort is distributed on Business Modeling and Requirements disciplines. During Elaboration phase, most effort is distributed on Requirements, Analysis Design and Implementation disciplines. In the Construction phase, most effort is distributed across Analysis Design, Implementation and Test disciplines. The following phase, the Transition phase, most effort is distributed across the Test and Deployment disciplines. Supporting Disciplines are distributed generally across the 4 phases.

Application of modeling standard, in particular the diagram of Unified Modeling Language (UML) is a significant part of the UP. Common modeling includes:

- Context model that shows how system fits to its overall environment
- High level business requirements model (usually refers to Use Case diagrams)
- Glossary defining critical business terms
- Domain model showing business classes or entities (Class Diagrams)
- Business process model that depicts high level overview of business processes supported by the system (Activity diagrams)

3.2.6 Comparison of Life-Cycle Models

Five different life-cycle models have been examined. The table (Table 3.1) below shows the comparison of strengths and weaknesses of these five models.

Table 3.1: Comparison of Life-cycle Models

LIFE-CYCLE MODELS	STRENGTHS	WEAKENESSES
Waterfall Model	• Disciplined Approach • Document Driven • Documentation ensures system requirements can be traced back to stated needs	• Delivered product may not meet client's needs • Documentation is expensive, time-consuming and hard to keep current • Client's needs may go unstated or misunderstood
Rapid-Prototyping Model	• User can try the system and provide constructive feedback during development • Ensures that delivered product meet client's needs • Enable early detection of errors and omissions	• Construction of prototype may delay schedule and budget overflow • Absent of or incomplete system documentation • System backup and recovery, performance, and security issues may be overlooked
Spiral Model	• Risk driven	• Used only for medium to large scale software production • Developers have to be competent in risk analysis and resolution
Interactive Multimedia Software Design and Development Model	• Focus on media integration • Suitable for handling and synchronizing media	• Suitable only for full multimedia development
Unified Process	• Closely models the real-world software production • For both small- and large-scale projects • Support for iteration and incrementation • Take advantage of Unified	• Has no particular standard for writing UML notation

	Modeling Language (UML) to model diagrams • Manage requirements well	

3.2.7 Chosen Life-Cycle Model

Unified Process (UP) is the chosen life-cycle model to build the Digital Game-based Learning of C++ Programming Language. It is the best object-oriented methodology available today and it provides the best solution to date for treating a large problem as a set of smaller, largely independent sub-problems.

Why I choose UP?

- The system I am building is a web-based system that focuses on games for learning purposes. Different user of web-based system requires different things provided in the website and what they can do in the website. Other than that, different people have different idea on what sort of games are suitable for learning C++. UP is an iterative and incremental methodology that encourages steps within a specific workflow to be repeated until requirements are satisfied.
- Any changes that the user request can also be catered using UP.
- The scope of this system is considered quite wide. Though it is stated that I am only concentrating on procedural programming in C++, the website providing learning must be taken into consideration as well. It is hard to focus on so many units of information at the same time and so, UP is a methodology that promotes development iteratively and incrementally.

- Furthermore, it is use-case driven and utilizes the Unified Modeling Language (UML). I have taken up a course on UML that explains briefly about modeling for UP. So, choosing UP may avoid further learning curve.

3.2.8 Relation of Unified Process with Proposed System

In the Inception phase of UP, the scope of the proposed system, Digital Game-based C++ Learning System is determined. The scope of the proposed system is C++ procedural programming. Other than that, in this phase, problem to be solved is identified as well. It is discovered that teaching C++ using traditional method or classroom method is boring, sometimes ineffective and the worst is, not motivating learner to learn C++ programming language. Three main actors were identified as well in this phase, namely Registered User, Visitor (both are generalized into User) and Administrator.

Elaboration phase is where requirements and architecture are established for the development of the proposed system. Functional requirements and non-functional requirements acquired through few elicitation techniques. Elicitation techniques include discussion, internet surfing, distributing questionnaires and collecting facts from existing software. Requirements collected are then analyzed and depicted in the form of Use Case Diagrams. From the Use Case Diagram, Class Diagram is generated and so are Sequence Diagrams and Activity Diagrams.

3.3 REQUIREMENTS ELICITATION

Requirements elicitation is a process of discovering requirements of the system. It is usually carried out by system developers and engineers together with customers and end-users. These people need to collect information on the application domain, problem to be solved, business which is buying the system and needs of system stakeholders. So, in other words, system developers and engineers are involved in 4 activities, namely, application domain understanding, problem understanding, business understanding and understanding the needs and constraints of system stakeholders. Figure 3-7 depicts the activities or components of requirements elicitation (Adapted from Kotonya's and Sommerville's book entitled, "Requirements Engineering: Techniques and Processes").

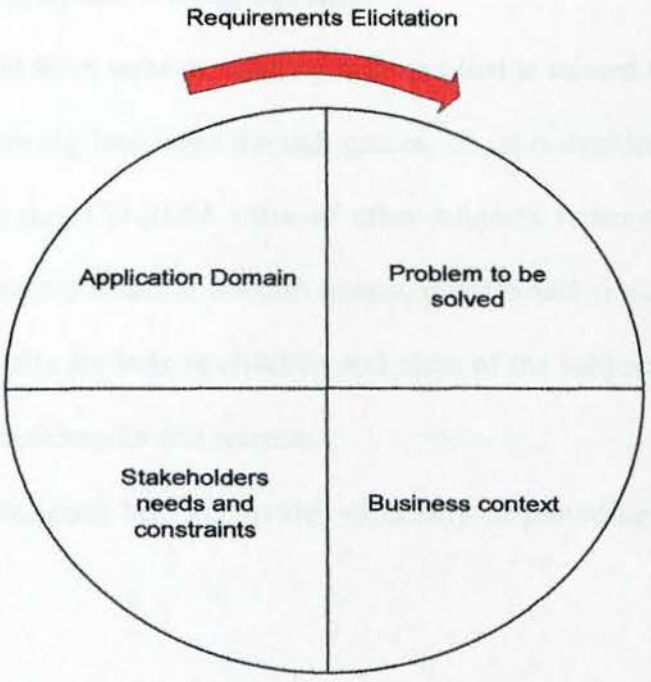


Figure 3.7: Requirements Elicitation Activities

Requirements elicitation is however, not a process of "fishing" for requirements. It is more to negotiation process involving all stakeholders to establish a set of requirements which are complete and consistent (Kotonya & Sommerville, 2003).

3.3.1 Research Methodology

Requirements elicitation involves discovering lots of information about the system. It is a cliché whereby several information discovery techniques about the problems to be solved, business context, application domain and needs of stakeholders are used. Described in the following sub-topics are the research methodologies or few elicitation techniques I have used to discover information on the software that I am developing.

3.3.1.1 Collecting facts from existing software

It is unfortunate that there were no existing software that is related to learning C++ or even other programming languages through games. So, it is decided that facts can be collected from learning CD-ROM titles of other subjects rather than programming languages that promotes learning through games, puzzles and simulations. Such CD-ROM titles are usually for kids or children and most of the subjects were focused on reading, spelling, mathematics and science.

From these software, much help is provided especially on providing ideas on the game development part.

3.3.1.2 Discussion

Discussion among friends especially those who are also developing game-based learning system helped a lot. Information regarding which appropriate languages or tool to use, which software model is suitable and et cetera is shared and debated among ourselves. Discussion also proves to be useful whenever I was unable to proceed or unsure of what to do next in the project. Friends provide their opinion and ideas in this matter.

Discussion with my facilitator, Pn. Siti Hafizah also helps a lot in relation on how to progress with my thesis; advices and direction of my report were also given.

3.3.1.3 Internet Surfing

Internet is a resource that I rely on most of the time. As books on game development and multimedia are sparse in the library, Internet becomes the next best choice I have. Definitions on terminologies, other people's researches, journals, articles and ideas are vast and easy to find through Internet surfing. It is very good as I get ideas, facts and anything related to my system from all over the world. However, the bad side of Internet is the reliability of the information I collected. To avoid that, I try to get information from reliable and approved journals I can possibly find through and within library sites.

3.3.1.4 Questionnaires

Questionnaires are prepared and given out to various people from various faculties in University of Malaya. These questionnaires focus on people's knowledge multimedia games education, also what are their perceptions of learning C++ through games and their requirements of digital game-based C++ learning system. Targeted people are students of my own faculty, Faculty of Computer Science and Information Technology, students from Engineering Faculty who are made compulsory to take up C++ programming as one of their core subject and students from other faculties who minor in Information Technology subject that require them to take up C++ programming. A sample of the questionnaire can found at Appendix A.

3.4 SUMMARY

This chapter captures the various development methodologies available today. Methodology selected is Unified Process, which is one of the best object-oriented methodologies that supports iteration and incremental of process that enables a system to be build in a more complete and elaborated environment.

Besides that, this chapter also explores elicitation techniques for requirements discovery. Research methodology or elicitation techniques that is used include collecting facts from existing software, discussions, internet surfing and questionnaires.

CHAPTER 4:

SYSTEM

ANALYSIS

CHAPTER 4: SYSTEM ANALYSIS

4.1 INTRODUCTION

In a system development process, after identifying the problem that needs to be solved and requirements are elicited, the next appropriate step is to analyze and understand the problem itself.

System analysis is a process that studies the problem thoroughly to deepen understanding towards the problem to be solved and the needs that triggered the software production. In this phase, requirements elicited are analyzed and if possible, identify solution requirements and expectations.

In a nutshell, system analysis involves:

- Understanding broader aspects of the software that would be required to solve the identified problem
- User's requirements to the software are carefully identified and documented. This includes functional requirements and non-functional requirements
- Address questions on how to achieve the user requirements in terms of component and interaction

4.2 ANALYSIS OF SURVEY CONDUCTED

4.2.1 Introduction

Survey had been carried out through questionnaire method. The information provided by respondents based on the returned questionnaires forms the sole basis for this system. The circulated questionnaire can be found in Appendix A.

4.2.2 Respondents Profile

The respondents chosen for this survey are those who are learning C++ programming language and also those who have learned C++ programming language. All respondents are University of Malaya's student and from various faculties such as Engineering Faculty, Science Faculty and not to forget, Faculty of Computer Science and Information Technology. The number of chosen respondents is 30.

4.2.3 Discussion of Results and Analysis

Questions that follows require the respondents to answer using scales from 1-5, very easy-very hard. The result is tabulated in Table 4.1

Table 4.1: Students Views on the Toughness of C++

Questions	No. of Respondents				
	Scale				
	1 Very Easy	2 Easy	3 Moderate	4 Hard	5 Very Hard
Q4. Rate the toughness for every sub-topic given below:					
a) Basic of programming		27	3		

b) Control Statements			20	10	
c) Functions			15	15	
d) Arrays				25	5
e) Pointers		2			28

Q.5 Which category of multimedia games is relevant t learn C++?

2% of the total respondents answered "Puzzle Games"

80 % of the total respondents answered "Word Games"

66% of the total respondents answered "Answer-and-Questions (Quiz-like game)"

71% of the total respondents answered "Adventure Games"

Q.6 For every type(s) of multimedia game(s), please give an example

For Puzzle Games, nobody suggested anything.

For Word Games, suggested games: Drag-and-drop games, hangman, code writing games.

For Adventure Games, suggested game: C++ embedded in games like treasure Hunt.

For Answer-and-Questions Games, suggested game: Quiz, Tutorial

Q.7 How would you like a game to help you?

93% of the respondents chose "Understand Definitions"

93%of the respondents chose "Understand Concept"

99% of the respondents chose "Write short codes or program"

3% of the respondents chose "Others", specifying "Write correct syntax"

Q.8 What kind of approach should be adopted by the game?

84% of the 30 respondents said that they prefer to be tutorials approach.

The remaining 16% prefers the notes approach

Q.9 What type of type of game do you expect to see?

20 out of 30 respondents, about 66% state they prefer combination of chapter games. The remaining respondents, 34% state they prefer combination of chapters.

Q.10 If it is a combination of chapters, how you would like the content of every chapter to be arranged in the game?

60% of the 10 respondents prefer that chapters are randomly arranged. The remaining 40% prefer chapters be sorted according to its level of toughness.

Q.11 How would you like to navigate through the game?

47% of 30 respondents said mouse only

53% of 30 respondents said combination of mouse and keyboard

Q.12 Other than games, what functions should be included in the website?

Notes: 95% of the total respondents

Sample source codes: 96% of the total respondents

Programming exercises: 70% of the total respondents

Links to relevant sites: 21% of the total respondents

Q.13 Should the website be assigned access levels (eg. member and non-member. Only member has the privilege to access the content of the website) ?

79% of the total respondents said yes.

The remaining 21% said no.

4.2.4 Conclusion of Questionnaire Analysis

Respondents provided good and positive input to the survey Games expected from the respondents are mostly word games, adventure games and answer-and-questions games. A game is expected to help in understanding definitions, concepts, write short codes and write proper syntax. Most respondents prefer games that are tutorial-like and provide questions based on a combination of chapters that are sorted randomly. However, there a number of people who prefer chapter specific games. Navigation through the game using a mixture of keyboard and mouse is preferred. Other than providing games in the web-based system, the user is also expecting to be provided with notes, sample source codes and programming exercises. Respondents also expect the website be assigned access levels so that the contents of the website are exclusively for registered users of the website.

4.3 FUNCTIONAL REQUIREMENTS

Functional requirement is a set of tasks that indicates what the system is required to perform. In other words, it describes the activities and services that a system must be able to provide to the user of the system. It shows an interaction between system that is being built with the environment. It also connotes how the system should behave under certain stimuli. Each functional requirement is independent of solutions to the problem to be solved.

Basically, there are three main actors in this proposed system, Digital Game-based C++ Learning System. Firstly, the administrator, secondly, the user of the system that can be generalized into registered user and visitor

The modules available in this system are:

1. Sign-in

   This is the module where registered users are required to login to access all contents provided by the website. Visitors, who would like to access other functionality in the site other than notes can register as well in this module.

2. Notes

   This module provides notes on C++ procedural programming. It basically covers 5 chapters: (1) Basics of programming; (2) Control statements; (3) Functions; (4) Arrays; and (5) Pointers.

3. Tutorials

Tutorial module is the module where registered user may download tutorial questions and answers. These tutorials serve as extra exercises for registered users. Both questions and answers will be in Word document.

4. Games

   Games module is where games for learning C++ programming is provided. Here the registered user may play the game online. Games such as hangman and puzzle games which are related to C++ programming are provided.

5. Code Dump

   This module is where source codes on various programming exercises are provided for the registered users to download. They may use it as a reference on how to write good programs, or even modify it to suite their programs.

6. Maintenance

   Maintenance module is only for the administrator. This module is where addition, deletion and updating to the contents of the website are done. This includes upload and deletion of notes, tutorials, games and source codes.

7. Help

   This module serves as a guide to users who have difficulties in how to use the website as their learning aid. It includes guides on how to download tutorials, games and source codes.

Shown in Figure 4.1 is a structure charting the functions of the proposed system (Digital Game-based C++ Learning System) together with the roles of each actor, namely administrator, registered user and visitor in the system.

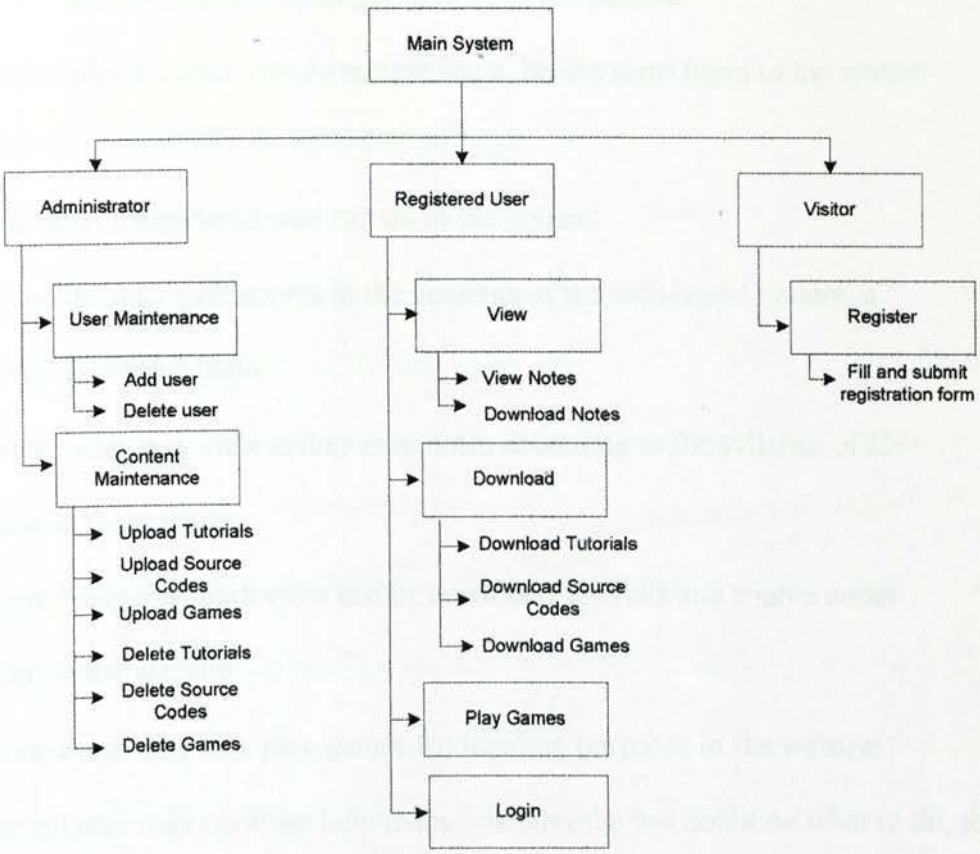


Figure 4.1: Structure Chart for Digital Game-based C++ Learning System

Based on Figure 4.1, it shows that the main system, that Digital Game-based C++ Learning System has 3 main actors. Actors are the administrator, registered user and visitor. Administrator is the person who maintains and owns the system Administrator has the responsibility to:

- Maintain content in the system itself. To maintain the content:

- Administrator may upload of notes, tutorials, games, source codes into the website.
- Administrator may delete notes, tutorials, games and source codes that are outdated or no longer in need in the website

- Before the administrator can do maintenance, he/she must login to the system

Following is what a registered user can do in the system:

- Before he/she may gain access to the contents of the web-based system, a registered user must login.
- Registered user may view and/or save notes according to the syllabus of C++ provided in the website
- Registered user may both view and/or download tutorials and source codes provided in the website
- Registered user also may play games for learning purposes in the website.
- Registered user may view the help menu when he/she has doubt on what to do, for example, how to download the contents of the website.
- Registered user may logout of the system when he/she feels like quitting the website.

A visitor generally refers to the normal user, is not allowed to access the website. He or she has to register himself/herself into the system by filling and submitting the registration form.

4.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are definitions of system properties and constraints under which the system must operate. It describes features included in the system to provide feasible and ease of use.

The non-functional requirements of this system are:

- User friendly
    - It uses graphical user interfaces which put the learner to ease when using and learning with the system
    - It also provide error messages or alert messages when error occurs
- Efficiency
    - Response time for interaction should be between 1 or 2 seconds
    - Time to download games especially, should be fast enough. However, it may depend on the network connection speed and memory constraints of the learner's computer itself
- Reliability
    - The website must always be available unless during maintenance
    - Must be also reliable enough to accurately manipulate data in and out of the database

4.5 TOOLS ANALYSIS RESULTS

4.5.1 System Platform

Platform chosen is Microsoft Windows. The reasons this operating system is chosen are, firstly, it is easy to install because it provides friendly graphical interface and wizard. Secondly and most important reason is that it supports web development tools better than other operating system.

The version of Microsoft Window that will be used is Windows XP Professional. Since I am developing a combination of web-based and multimedia system, therefore, usage of memory will be high. Windows XP provides better memory handling compared to older versions of Windows. Other than that, it integrates Internet Explorer into the graphical user interface. Furthermore, it has higher stability, USB functions and a satisfactory array of driver support for hardware.

4.5.2 Programming Languages and Technologies

Generally, I will be using a combination of several programming languages for the development of different modules, namely ActionScript for development of games, JavaScript, ASP and HTML for web development.

ActionScript is actually a language used in Macromedia Flash. Macromedia Flash has interesting features and a powerful programming language, ActionScript that enables the development of interactive, multimedia incorporated games. This is the reason why ActionScript is chosen.

ASP and HTML will be used for web programming. It is most suitable for web design, building and maintenance. Software for web design, building and maintenance is Macromedia Dreamweaver.

4.5.3 Web Application Development Tools

4.5.3.1 Image Creation and Editing Tools

Adobe Photoshop Elements will be the choice for image creation and editing. This is because it is the best image editor and manipulator for novice users, like me. It provides a vast range basic function and manipulation tools that are easy to use and understand.

4.5.3.2 Digital Sound Editor, Application and Tools

Audacity is chosen because it is a free program that can be downloaded from the Internet. Furthermore, it has all the basic function needed for basic sound editing. Its usage is almost similar to those expensive sound editing software such as Sonar though less comprehensive.

4.5.3.3 Animation tool

Macromedia Flash MX 2004 will be used as the application for generating animation effects. The reason this software is selected because Flash movies are small and compact and can be downloaded rapidly. Downloading time is especially important in a web-based environment. Furthermore, Flash MX 2004 provides better web handling features than the previous Flash versions. It supports ActionScript 2.0 which is more

comprehensive and powerful than the previous ActionScript 1.0. In addition, it has a feature that protects the Flash movie to be copyrighted. This is especially important to preserve the integrity of the games.

4.5.3.4 Authoring tool

Again, Macromedia Flash MX 2004 will be the authoring tool for compiling all elements of media for the game. Flash MX 2004 is a greater tool for authoring multimedia application that is meant for the web.

4.5.3.5 Web page Design Tool

Macromedia Dreamweaver MX 2004 will be used for web page design, building and maintanence. Macromedia Dreamweaver MX 2004 is friendly, easy to use, and provides a wide range of development tools. It enables the building of both ASP and HTML pages.

4.5.4 Web Server Software

Internet Information System (IIS) is selected instead of Personal Web Server (PWS) due to the fact that Windows XP Professional, the chosen system platform has no support for PWS.

4.5.5 Web Browser

Basically, Internet Explorer 6.0 will be the selected web browser. It is the default web browser and it comes with the system platform chosen earlier (Section 4.5.1), Windows XP Professional.

4.5.6 Database System

Database system chosen is Microsoft Access 2002. I am more familiar with this database system. It is easier to maintain and it even provides security features.

4.5.7 Summary of Tools Analysis

Tools that are to be used for development are:

- System Platform : Microsoft Windows XP Professional
- Programming Language : ActionScript, ASP, HTML
- Image creation and editing tool : Adobe Photoshop Elements
- Sound editing tool : Audacity
- Animation tool : Macromedia Flash MX 2004
- Authoring tool : Macromedia Flash MX 2004
- Web-page development tool : Macromedia Dreamweaver MX 2004
- Web Server Software : Internet Information System (IIS) 5.0
- Web Browser : Internet Explorer 6.0
- Database : Microsoft Access 2002

4.6 SYSTEM REQUIREMENTS

4.6.1 Development Environment

4.6.1.1 Hardware Requirements

The hardware requirements are:

- PC with at least Pentium III pro
- 128MB RAM
- 8GB of Hard Disk Memory
- 15" monitor, 16-bit colour screen
- Keyboard
- Mouse

4.6.1.2 Software Requirements

The software requirements are:

- Adobe Photoshop Elements
- Audacity
- Macromedia Flash MX 2004
- Macromedia Dreamweaver MX 2004
- Microsoft Access 2002
- Internet Explorer 6.0

4.6.2 Runtime Environment

4.6.2.1 Hardware Requirements

The hardware requirements are:

- PC with at least Pentium III pro
- 128MB RAM
- 8GB of Hard Disk Memory
- 15" monitor, 16-bit colour screen
- Keyboard
- Mouse

4.6.2.2 Software Requirements

The software requirements are:

- Windows 98 and above
- Internet Explorer 6.0
- Macromedia Flash 7.0 plug-in

4.7 SUMMARY

This chapter, on the whole, focused on detailed examination of the survey carried out to find out the requirements of various users on the system to be built, software and tools to be used for the development of the system and system requirements to ensure the development is smooth.

CHAPTER 5:

SYSTEM DESIGN

CHAPTER 5: SYSTEM DESIGN

5.1 INTRODUCTION

System design is a process whereby specifications from the system analysis process are put into production. System design should:

- Implement all explicit requirement contained in the analysis and accommodate all the implicit requirements desired
- Provide complete picture of the system, address the data, functional and behavioural domains from an implementation perspectives

5.2 SYSTEM ARCHITECTURE

Digital Game-based C++ Learning System is a web-based learning system. It means that those who access the website do not need to install the system into their personal computers. This is especially useful and easier because people accessing the website are not bound to use it from a specific computer. Most of the contents of the proposed system such as tutorials, notes, source codes and games needed to be downloaded every time the learner wants to use it. This is not a huge problem as most of the content are lightweight. The only thing needed by the learner is Internet connection.

The proposed web-based system is a web application. So, the system's website has a three-tier architecture. The three-tier architecture divides the system into 3 layers, namely the thin client (web browser), middleware (servers) and storage layer (database management system). The architecture is depicted in Figure 5.1.

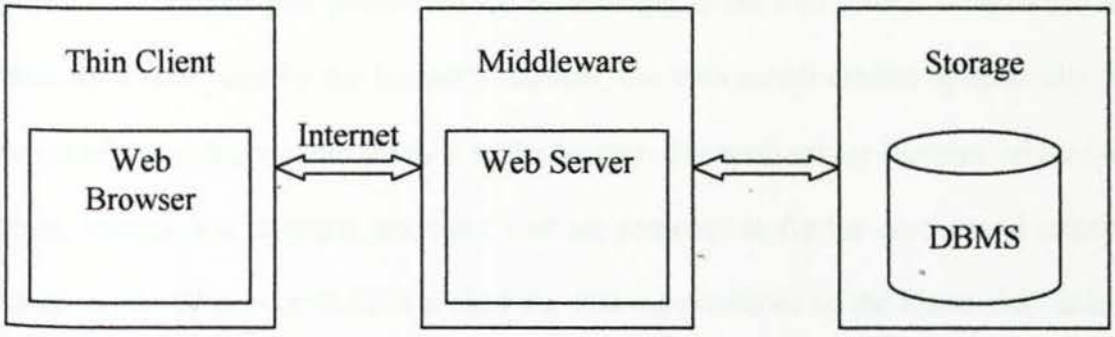


Figure 5.1: Three-Tier Architecture

5.2.1 Thin Client

'Thin Client' is where the web browser resides. Web browser facilitates user interaction between learner with the system through user interfaces. Common browsers such as Internet Explorer and Netscape are usually used by learner to browse pages of the system's website. Macromedia's Flash Player must be installed in the browser as plug-in, since the games included in the game section are Flash movies. In a nutshell, the thin client is only a piece of software running in the learners' personal computers. It does not contain the application logic and does not use the local hard drive for storing information.

5.2.2 Middleware

Middle tier contains the web server software such as Internet Information System (IIS) or Personal Web Server (PWS) that is responsible for providing functionality to the website. The middleware, at the same time handles the communications, presents the contents of the system to the users and connects the client software transparently with the storage system.

The main component that constitutes the middleware is the web server. When a request comes for a web page by the learner's browser, the web server creates dynamically the content of the web page and sends it to the learner. The web server contains server-side scripts, written in a scripting language, that are responsible for the on-demand creation of web pages. Wherever there is a need for making decisions in the client-side, client-side JavaScript code is utilized. The web server also contains all other files that are needed, such as Flash files and images. The scripts in the server connect to the database management system in order to obtain the needed information or to store information. When there is a need to store learner's information, the thin client sends the information to the web browser, the Web server passes this information to the relevant script and this script stores the information in the storage system.

5.2.3 Storage Layer

Storage layer consists of the database management system. The proposed Digital Game-based C++ Learning System is using a relational database management system. Its main function is to store data. The thin client does not have direct access to the storage system. Instead, the scripts of the server are the only software module that accesses the database management system.

Systems can be viewed from a number of perspectives. Different people such as the user of a system and the people who are actually doing the programming of the system viewed the system from dissimilar viewpoints. System architecture is used to manage

these different viewpoints, therefore controlling the development of the system and its life-cycle.

From the previous chapters, it is stated that the methodology used for the development of Digital Game-based C++ Learning System is Unified Process. Unified Process takes advantage of Unified Modeling Language (UML) which captures all different viewpoints as a set of 5 interlocking views as shown in Figure 5.2

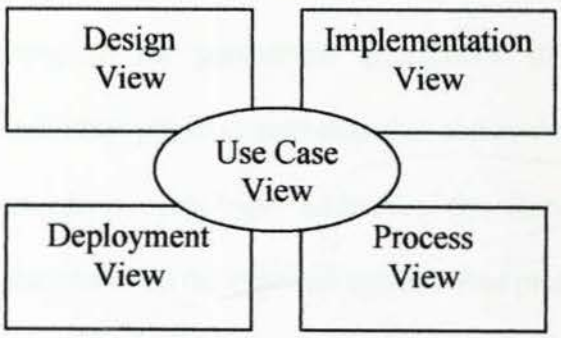


Figure 5.2: Five Architectural Views

Use Case View encompasses on scenarios as viewed by human users and external system. It expresses in general what the new system will do without specifying how. The end product of use case view is the use case diagram.

Design View focuses and support functional requirements of the system. It encompasses classes, interfaces and collaboration that in the end form the vocabulary of the problem and solution. Design view is captured in class and object diagrams which forms the static view of the system and activity and statechart diagram which forms the dynamic view of the system.

Process View focuses on aspects of the system that involves time and flow of control. It addresses the performance, scalability and throughput of the system. Process view is captured in active class diagrams, activity diagrams, and sequence diagrams.

Implementation View encompasses the structural and behavioural dimensions of the system. It focuses on the components and files used to assemble and release the physical system. The end-product of the implementation view is the component diagram.

Deployment View focuses on the geographic distribution of the various software elements on hardware and other physical elements that forms the system. It consists of nodes that forms the hardware topology, addresses the distribution, delivery and installation of the parts that make up the physical system. End product of the deployment view is none other than the deployment diagram.

5.3 USE CASE VIEW OF THE PROPOSED SYSTEM

Figure 5.3 shows the overview of the Digital Game-based C++ Learning System. The system consists of several packages. Packages are used to group things or classes. It is used arrange your modeling elements into large chunks that is easier to manipulate as a group. The packages included in this system are:

- Notes
- Tutorials
- Game
- Source Codes

- Maintenance
- Sign-In
- Help

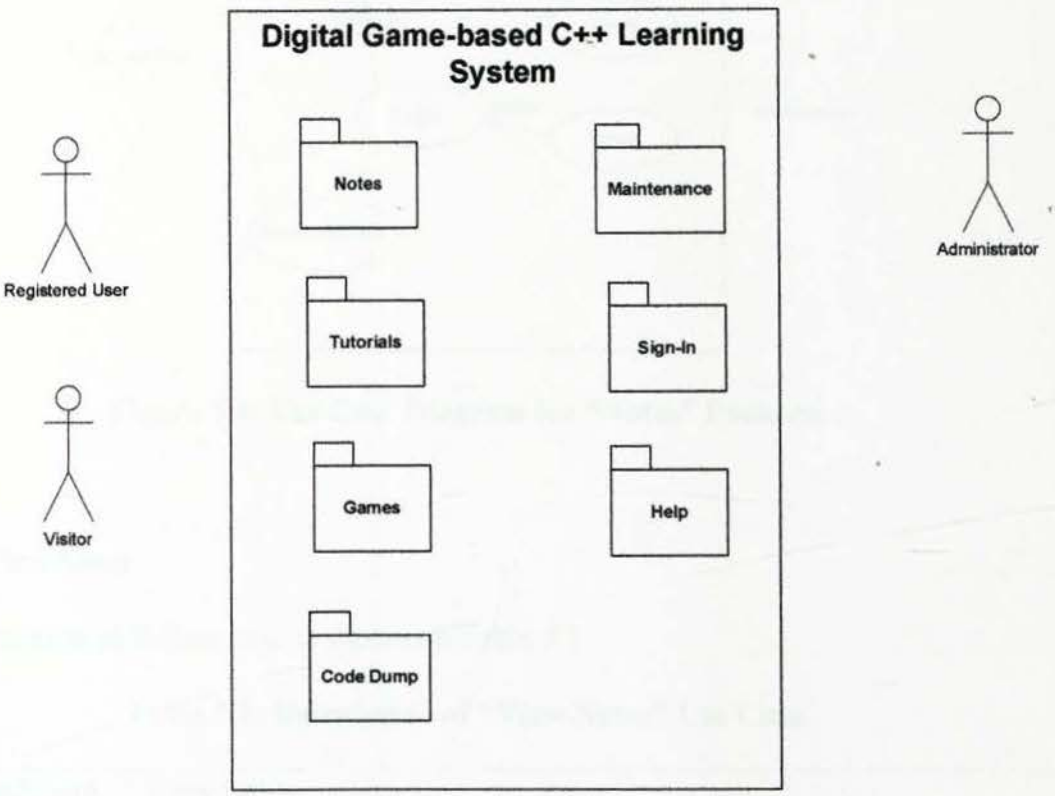


Figure 5.3: Overview of the Digital Game-based C++ Learning System

Note:

In some of the description of a particular use case, the term "User" is sometimes used when it involves more than 1 actor. For example, if a registered user and a visitor have the use case, both the registered user and visitor will be referred to as "User" in general.

5.3.1 "Notes" Package Use Case Diagram

Table 5.4 depicts the use case diagram for package "Notes".

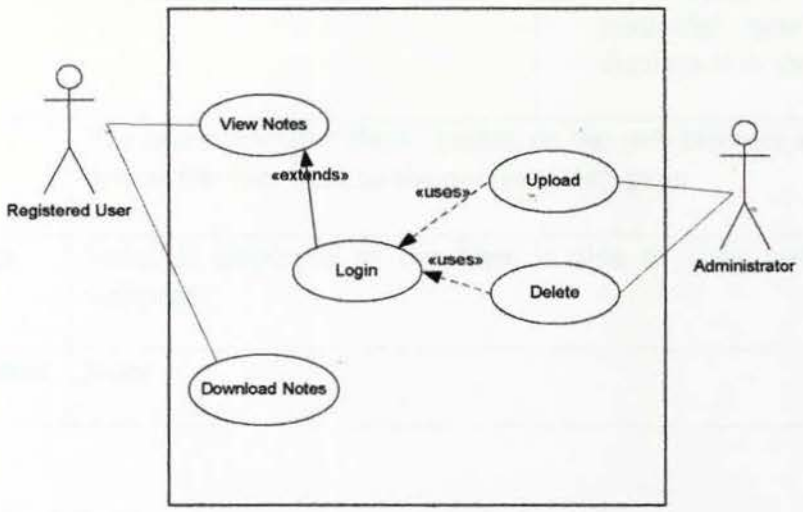


Figure 5.4: Use Case Diagram for "Notes" Package

5.3.1.1 View Notes

The description of the use case is shown on Table 5.1.

Table 5.1: Description of "View Notes" Use Case

Use Case Name	View Notes	
Priority	Desired	
Participating Actors	Registered User, and Administrator	
Description	This use case describes the event of a user (Registered user or visitor) viewing notes provided at the website.	
Precondition	Registered user must login before viewing notes	
Trigger	This use case is initiated when the user selects a particular chapter to be viewed in the "Notes" page.	
Basic Path	**Actor Action**	**System Response**
	1. User selects "Notes" option from the homepage.	2. The system responded by displaying the Notes webpage. Notes webpage contains a list of chapters

	3. User selects a topic	4. The system retrieves the particular note chapter and displays it to the user.
Alternative Paths	The user clicks on "Back" button on the web browser and the system directs the user back to the previous web page	
Postcondition	Notes is displayed to the User is able to view notes within the webpage.	
Exception Paths	None	

5.3.1.2 Download Notes

Description of the use case is shown on Table 5.2

Table 5.2: Description of "Download Notes" Use Case

Use Case Name	Download Notes	
Priority	Desired	
Participating Actors	Registered User	
Description	This use case describes the event of a registered user downloading notes provided in the website	
Precondition	The registered must have performed login before downloading can be done	
Trigger	This use case is initiated when the registered user selects to notes downloading option on the website	
Basic Path	**Actor Action**	**System Response**
	1. Registered user clicks on "Notes" button in the homepage	2. The system responded by displaying the Notes webpage.
	3. Registered user selects note to be downloaded from the uploaded tutorial list	4. System responded by starting the downloading process by retrieving notes from the database.

Alternative Paths	The registered user clicks on "Back" button on the web browser and the system directs the registered user back to the previous web page	
Postcondition	Note(s) is downloaded from the webpage.	
Exception Paths	None	

5.3.1.3 Upload Notes

Description of the use case is shown on Table 5.3.

Table 5.3: Description of "Upload Notes" Use Case

Use Case Name	Upload Notes	
Priority	Desired	
Participating Actors	Administrator	
Description	This use case describes the event of an administrator adding new notes to the website	
Precondition	The administrator must have performed login before uploading can be done	
Trigger	This use case is initiated when the administrator selects to perform maintenance on the website	
Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.
	3. Administrator selects "Notes" option from the maintenance list	4. System responded by displaying the Notes Maintenance Page
	5. Administrator enters in the note's information and then browse for the note to be uploaded	7. System uploads the note and

	6. User clicks "Upload" button. 9. Administrator may continue uploading	stores the record in the database 8. System notify the administrator that the file was uploaded
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Note(s) is added to the webpage.	
Exception Paths	None	

5.3.1.4 Delete Notes

Description of the use case is shown on Table 5.4.

Table 5.4: Description of "Delete Notes" Use Case

Use Case Name	Delete Notes	
Priority	Desired	
Participating Actors	Administrator	
Description	This use case describes the event of an administrator deleting outdated or no longer in use notes in the website	
Precondition	The administrator must have performed login before uploading can be done	
Trigger	This use case is initiated when the administrator selects to perform maintenance on the website	
Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.

	3. Administrator selects "Notes" option from the maintenance list	4. System responded by displaying the Notes Maintenance Page and a list of uploaded notes
	5. Administrator select note(s) to be deleted	
	6. Administrator clicks "Delete" button.	7. System prompts for deletion confirmation.
	8. Administrator confirms deletion by clicking "OK"	8. System removes the record(s) and file(s) for selected note(s) from the database.
	9. Administrator may continue deletion	
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Note(s) is deleted from the webpage.	
Exception Paths	None	

5.3.2 "Tutorials" Package Use Case Diagram

The figure below, Figure 5.5 depicts the use case diagram for the "Tutorials" package.

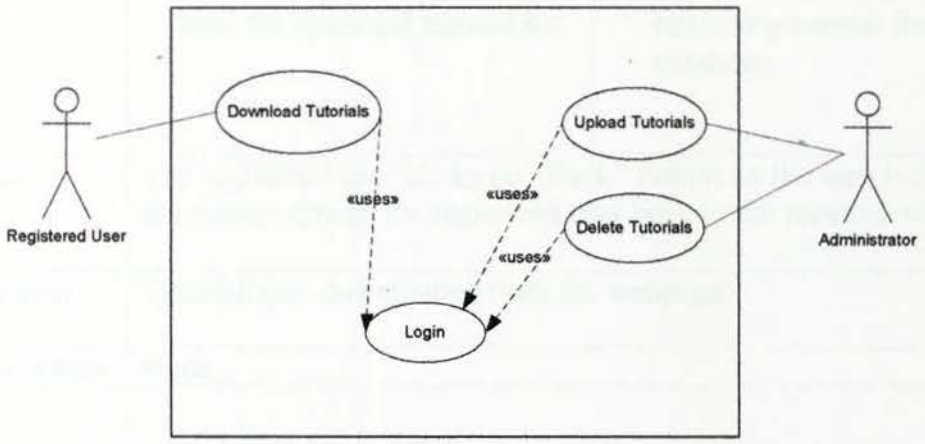


Figure 5.5: Use Case Diagram for "Tutorials" Package

5.3.2.1 Download Tutorials

Description of the use case is shown on Table 5.5.

Table 5.5: Description of "Download Tutorials" Use Case

Use Case Name	Download Tutorials	
Priority	Desired	
Participating Actors	Registered User	
Description	This use case describes the event of a registered user downloading tutorials provided in the website	
Precondition	The registered must have performed login before downloading can be done	
Trigger	This use case is initiated when the registered user selects to tutorials downloading option on the website	
Basic Path	**Actor Action**	**System Response**
	1. Registered user clicks on "Tutorials" button in the	2. The system responded by displaying the Tutorials

	homepage 3. Registered user selects tutorials to be downloaded from the uploaded tutorial list	webpage. 4. System responded by starting the downloading process by retrieving tutorial from the database.
Alternative Paths	The registered user clicks on "Back" button on the web browser and the system directs the registered user back to the previous web page	
Postcondition	Tutorial(s) is downloaded from the webpage.	
Exception Paths	None	

5.3.2.2 Upload Tutorials

Description of the use case is shown on Table 5.6.

Table 5.6: Description of "Upload Tutorials" Use Case

Use Case Name	Upload Tutorials	
Priority	Desired	
Participating Actors	Administrator	
Description	This use case describes the event of an administrator adding new tutorials to the website	
Precondition	The administrator must have performed login before uploading can be done	
Trigger	This use case is initiated when the administrator selects to perform maintenance on the website	
Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.
	3. Administrator selects "Tutorials" option from the maintenance list	4. System responded by displaying the Tutorials Maintenance Page

	5. Administrator enters in the tutorial's information and then browse for the tutorial to be uploaded 6. User clicks "Upload" button. 9. Administrator may continue uploading	7. System uploads the tutorial and stores the record in the database 8. System notify the administrator that the file was uploaded
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Tutorial(s) is added to the webpage.	
Exception Paths	None	

5.3.2.3 Delete Tutorials

Description of the use case is shown on Table 5.7.

Table 5.7: Description of "Delete Tutorials Use Case

Use Case Name	Delete Tutorials
Priority	Desired
Participating Actors	Administrator
Description	This use case describes the event of an administrator deleting outdated or no longer in use tutorials in the website
Precondition	The administrator must have performed login before uploading can be done
Trigger	This use case is initiated when the administrator selects to perform maintenance on the website

Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.
	3. Administrator selects "Tutorials" option from the maintenance list	4. System responded by displaying the Tutorials Maintenance Page and a list of uploaded notes
	5. Administrator select tutorials(s) to be deleted	
	6. Administrator clicks "Delete" button.	7. System prompts for deletion confirmation.
	8. Administrator confirms deletion by clicking "OK"	8. System removes the record(s) and file(s) for selected tutorials(s) from the database.
	9. Administrator may continue deletion	
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Tutorial(s) is deleted from the webpage.	
Exception Paths	None	

5.3.3 "Games" Package Use Case Diagram

Figure 5.6 depicts the use case diagram for "Games" package.

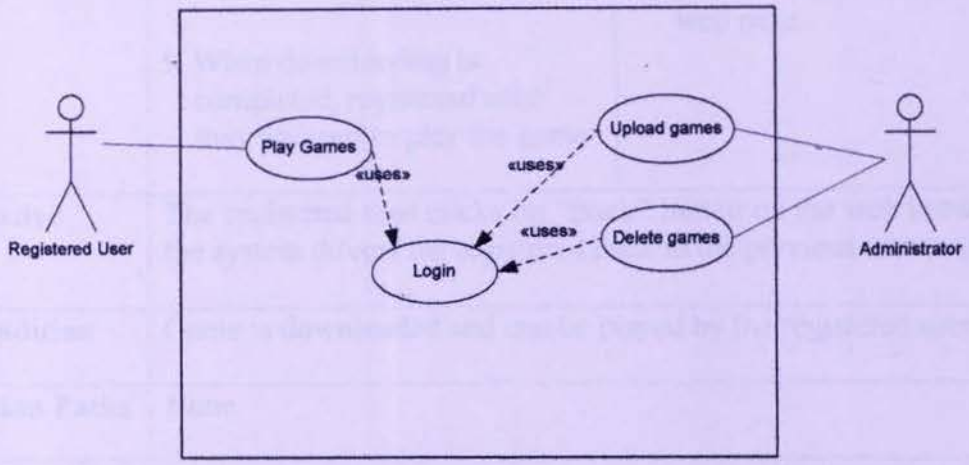


Figure 5.6: Use Case Diagram for "Games" Package

5.3.3.1 Play Games

Description of the use case is shown on Table 5.8.

Table 5.8: Description of "Play Games" Use Case

Use Case Name	Play Games	
Priority	Desired	
Participating Actors	Registered User	
Description	This use case describes the event of a registered user accessing games in the website	
Precondition	The registered user must be logged in before games can be accessed	
Trigger	This use case is initiated when the registered user selects the option to access games on the website	
Basic Path	**Actor Action**	**System Response**
	1. Registered user clicks on "Games" button	2. The system responded by displaying the Games webpage with a list of games

	3. Register user selects a game title from the list of games 5. When downloading is completed, registered user may proceed to play the game	4. System responded by downloading and opening the particular game in another web page
Alternative Paths	The registered user clicks on "Back" button on the web browser and the system directs the registered back to the previous web page	
Postcondition	Game is downloaded and can be played by the registered user	
Exception Paths	None	

5.3.3.2 Upload Games

Description of the use case is shown on Table 5.9.

Table 5.9: Description of "Upload Games" Use Case

Use Case Name	Upload Games	
Priority	Desired	
Participating Actors	Administrator	
Description	This use case describes the event of an administrator adding new games to the website	
Precondition	The administrator must have performed login before uploading can be done	
Trigger	This use case is initiated when the administrator selects to perform maintenance on the website	
Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.
	3. Administrator selects	4. System responded by

	"Games" option from the maintenance list 5. Administrator enters in the game's information and then browse for the tutorial to be uploaded 6. User clicks "Upload" button. 9. Administrator may continue uploading	displaying the Games Maintenance Page 7. System uploads the game and stores the record in the database 8. System notify the administrator that the file was uploaded
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Game(s) is added to the webpage.	
Exception Paths	None	

5.3.3.3 Delete Games

Description of the use case is shown on Table 5.10.

Table 5.10: Description of "Delete Games" Use Case

Use Case Name	Delete Games
Priority	Desired
Participating Actors	Administrator
Description	This use case describes the event of an administrator deleting outdated or no longer in use games in the website
Precondition	The administrator must have performed login before uploading can be done

Trigger	This use case is initiated when the administrator selects to perform maintenance on the website	
Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.
	3. Administrator selects "Games" option from the maintenance list	4. System responded by displaying the Games Maintenance Page and a list of uploaded notes
	5. Administrator select games(s) to be deleted	
	6. Administrator clicks "Delete" button.	7. System prompts for deletion confirmation.
	8. Administrator confirms deletion by clicking "OK"	8. System removes the record(s) and file(s) for selected game(s) from the database.
	9. Administrator may continue deletion	
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Game(s) is deleted from the webpage.	
Exception Paths	None	

5.3.4 "Code Dump" Package Use Case Diagram

The figure, Figure 5.7 shows the Code Dump Use Case Diagram of Code Dump package.

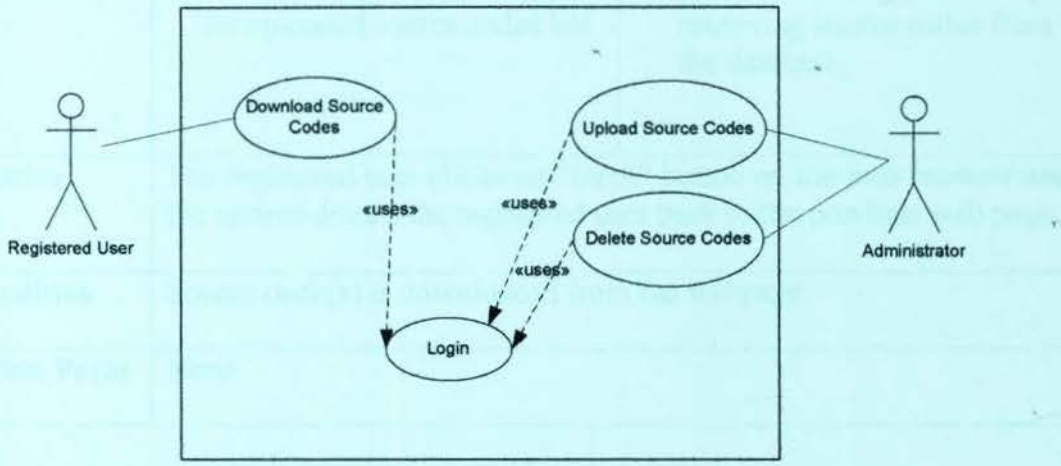


Figure 5.7: Use Case Diagram for "Code Dump" Package

5.3.4.1 Download Source Codes

Description of the use case is shown on Table 5.11.

Table 5.11: Description of "Download Source Codes" Use Case

Use Case Name	Download Source Codes	
Priority	Desired	
Participating Actors	Registered User	
Description	This use case describes the event of a registered user downloading source codes provided in the website	
Precondition	The registered must have performed login before downloading can be done	
Trigger	This use case is initiated when the registered user selects to source codes downloading option on the website	
Basic Path	**Actor Action**	**System Response**

	1. Registered user clicks on "Code Dump" button in the homepage 3. Registered user selects source codes to be downloaded from the uploaded source codes list	2. The system responded by displaying the Code Dump webpage. 4. System responded by starting the downloading process by retrieving source codes from the database.
Alternative Paths	The registered user clicks on "Back" button on the web browser and the system directs the registered user back to the previous web page.	
Postcondition	Source code(s) is downloaded from the webpage.	
Exception Paths	None	

5.3.4.2 Upload Source Codes

Description of the use case is shown on Table 5.12.

Table 5.12: Description of "Upload Source Codes" Use Case

Use Case Name	Upload Source Codes	
Priority	Desired	
Participating Actors	Administrator	
Description	This use case describes the event of an administrator adding new source codes to the website	
Precondition	The administrator must have performed login before uploading can be done	
Trigger	This use case is initiated when the administrator selects to perform maintenance on the website	
Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.

	3. Administrator selects "Code Dump" option from the maintenance list 5. Administrator enters in the source code's information and then browse for the source code to be uploaded 6. User clicks "Upload" button. 9. Administrator may continue uploading	4. System responded by displaying the Source Code Maintenance Page 7. System uploads the source code and stores the record in the database 8. System notify the administrator that the file was uploaded
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Source code(s) is added to the webpage.	
Exception Paths	None	

5.3.4.3 Delete Source Codes

Description of the use case is shown on Table 5.13.

Table 5.13: Description of "Delete Source Codes" Use Case

Use Case Name	Delete Source Codes
Priority	Desired
Participating Actors	Administrator
Description	This use case describes the event of an administrator deleting outdated or no longer in use source codes in the website
Precondition	The administrator must have performed login before uploading can

	be done	
Trigger	This use case is initiated when the administrator selects to perform maintenance on the website	
Basic Path	**Actor Action**	**System Response**
	1. Administrator clicks on "Maintenance" button	2. The system responded by displaying the Maintenance webpage.
	3. Administrator selects "Code Dump" option from the maintenance list	4. System responded by displaying the Code Dump Maintenance Page and a list of uploaded notes
	5. Administrator select source code(s) to be deleted	
	6. Administrator clicks "Delete" button.	7. System prompts for deletion confirmation.
	8. Administrator confirms deletion by clicking "OK"	8. System removes the record(s) and file(s) for selected source code(s) from the database.
	9. Administrator may continue deletion	
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the administrator back to the previous web page	
Postcondition	Source code(s) is deleted from the webpage.	
Exception Paths	None	

5.3.5 "Sign-In" Package Use Case Diagram

The overall use case diagram for "Sign-In" package is shown in Figure 5.8.

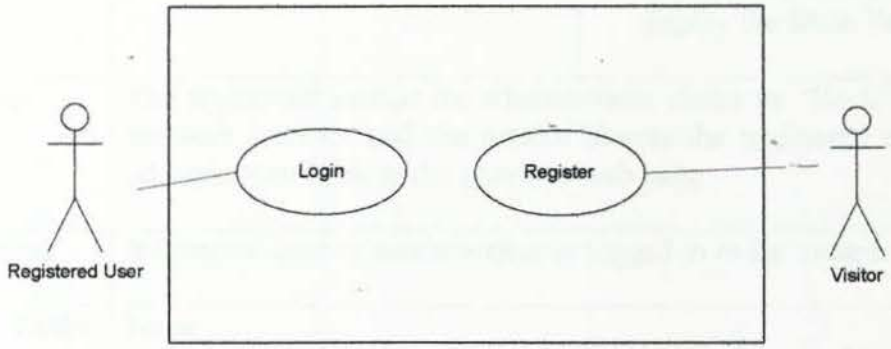


Figure 5.8: Use Case Diagram for "Sign-In" Package

5.3.5.1 Login

Description of the use case is shown on Table 5.14.

Table 5.14: Description of "Login" Use Case

Use Case Name	Login	
Priority	Desired	
Participating Actors	Administrator	
Description	This use case describes the event of an administrator or a registered signing in to the website to access all the contents of the system.	
Precondition	The administrator or the registered user must be registered with the system	
Trigger	This use case is initiated when the administrator or registered user selects the option to login	
Basic Path	**Actor Action**	**System Response**
	1. Registered user or administrator enters their userID and their password	
	2. Registered user or	3. System verifies the userID

	administrator click "Login"	and password against data in the database. If it is valid, the system will display the Main Page/
Alternative Paths	The registered user or the administrator clicks on "Back" button on the web browser and the system directs the registered user or the administrator back to the previous web page	
Postcondition	Registered user or administrator is logged in to the system	
Exception Paths	None	

5.3.5.2 Register

Description of the use case is shown on Table 5.15.

Table 5.15: Description of "Register" Use Case

Use Case Name	Register	
Priority	Desired	
Participating Actors	Visitor	
Description	This use case describes the event of a visitor registering to the system	
Precondition	No precondition	
Trigger	This use case is initiated when the visitor decides to register to the system to access all the contents in the website	
Basic Path	**Actor Action**	**System Response**
	1. Visitor clicks on "New User?" hyperlink	2. The system responded by displaying the registration form to the visitor
	5. Visitor fills in his/her information appropriately. Upon completion, click "Submit"	6. System validates the registration form so that all required fill must not be empty.

		If valid, the system will update its database and generate registration confirmation to the user.
Alternative Paths	The visitor clicks on "Back" button on the web browser and the system directs the visitor back to the previous web page	
Postcondition	Visitor is registered to the system. Visitor is now a registered user.	
Exception Paths	None	

5.3.6 "Help" Package Use Case Diagram

Figure 5.9 depicts the use case diagram for "Help" package.

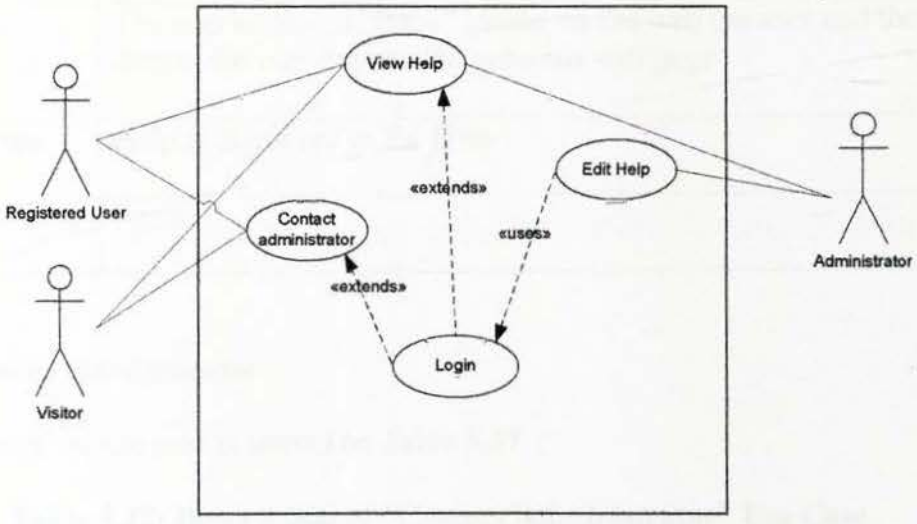


Figure 5.9: Use Case Diagram for "Help" Package

5.3.6.1 View Help

Description of the use case is shown on Table 5.16.

Table 5.16: Description of "View Help" Use Case

Use Case Name	View Help

Priority	Desired	
Participating Actors	Registered User, Visitor and Administrator	
Description	This use case describes the event of a user (Registered user or visitor) viewing help menu provided at the website.	
Precondition	No precondition. Both registered user and visitor can view the help with or without logging in.	
Trigger	This use case is initiated when the user selects the "Help" option in the homepage	
Basic Path	**Actor Action**	**System Response**
	1. User selects "Help" option from the homepage.	2. The system responded by displaying the Help webpage.
Alternative Paths	The user clicks on "Back" button on the web browser and the system directs the user back to the previous web page	
Postcondition	Help is displayed to the User	
Exception Paths	None	

5.3.6.2 Contact Administrator

Description of the use case is shown on Table 5.17

Table 5.17: Description of "Contact Administrator" Use Case

Use Case Name	Contact Administrator
Priority	Desired
Participating Actors	Registered User, Visitor
Description	This use case describes the event of a user (Registered user or visitor) making queries by contacting the administrator through e-mail
Precondition	No precondition. Both registered user and visitor can view the help with or without logging in.

Trigger	This use case is initiated when the user selects the option to make queries by sending the administrator an e-mail	
Basic Path	**Actor Action**	**System Response**
	1. User selects "Help" option from the homepage. 3. User is unable to find answers to the questions he/she wants answers to. 4. User clicks on the "Contact Administrator" hyperlink. 6. User composes the queries and click "Send" to send the e-mail directly to the administrator's inbox. 7. Administrator will reply the queries to the user's inbox	2. The system responded by displaying the Help webpage. 5. The system responded by opening a new window for composing e-mail.
Alternative Paths	The user clicks on "Back" button on the web browser and the system directs the user back to the previous web page	
Postcondition	Queries made through email	
Exception Paths	None	

5.3.6.3 Edit Help

Description of the use case is shown on Table 5.18.

Table 5.18: Description of "Edit Help" Use Case

Use Case Name	Edit Help
Priority	Desired
Participating Actors	Administrator

Description	This use case describes the event of an administrator updating help webpage.	
Precondition	Administrator must be logged in to the website	
Trigger	This use case is initiated when the administrator selects the help maintenance option.	
Basic Path	**Actor Action**	**System Response**
	1. Administrator selects "Help" option from the homepage. 3. Administrator updates the help menu and saves the changes.	2. The system responded by displaying the Help webpage. Notes webpage contains a list of questions and answers for troubleshooting the system
Alternative Paths	The administrator clicks on "Back" button on the web browser and the system directs the user back to the previous web page	
Postcondition	Help webpage is updated	
Exception Paths	None	

5.4 DESIGN VIEW OF THE PROPOSED SYSTEM

Figure 5.10 depicts the class diagram with classes and relationships that exist in the proposed system.

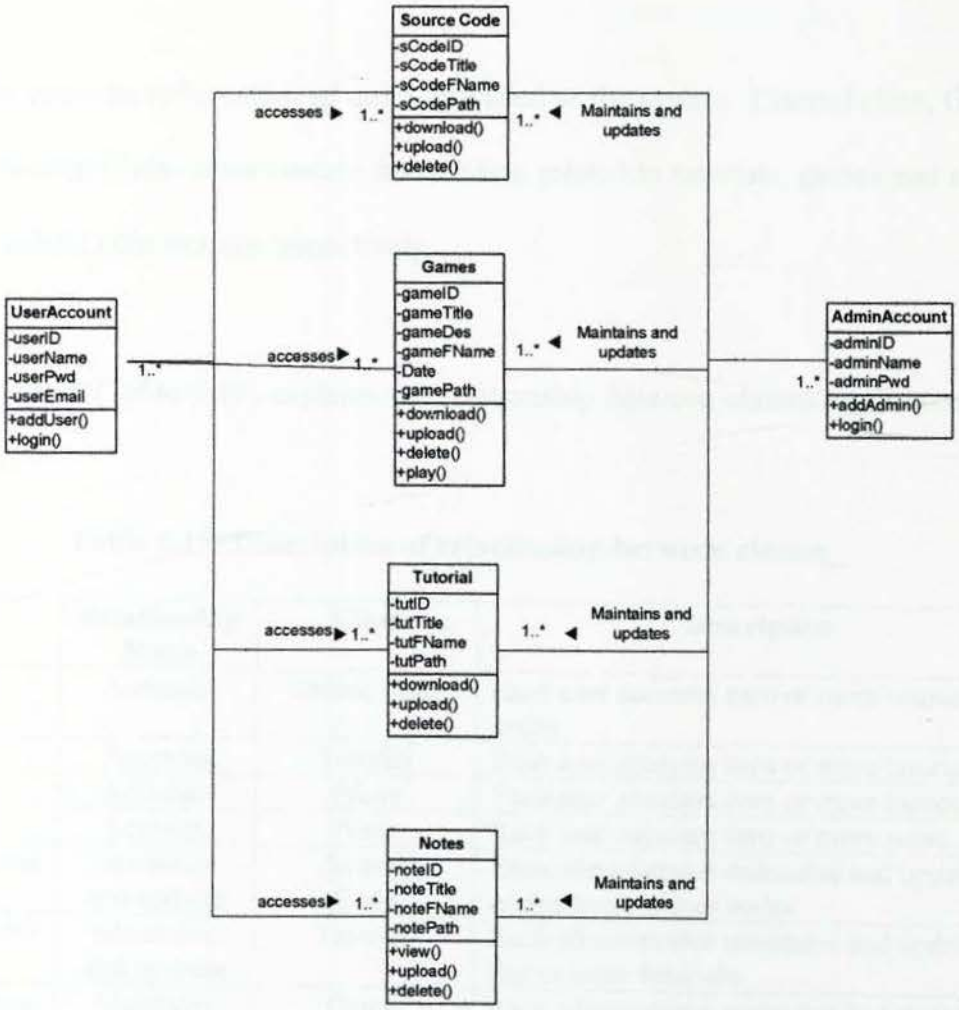


Figure 5.10: Class Diagram for Digital Game-based C++ Learning System

There are 6 main classes and they are the User class, Administrator class, Note class, Tutorial class, Game class and Source Code class.

User class contains all details of the registered users. It includes important attributes such as userID and password for signing-in purposes.

Administrator class contains all details of administrator(s). It includes important attributes such as adminID and password for signing-purposes for maintenance.

Notes class contains information of notes uploaded to the system. Tutorial class, Games class and Source Codes class contain information related to tutorials, games and source codes uploaded to the website respectively.

The table below (Table 5.19) explains the relationship between classes as illustrated in Figure 5.10.

Table 5.19: Description of relationship between classes

Class	Relationship Name	Class	Description
User	Accesses	Source Code	Each user accesses zero or more source codes
User	Accesses	Tutorial	Each user accesses zero or more tutorials
User	Accesses	Game	Each user accesses zero or more games
User	Accesses	Note	Each user accesses zero or more notes
Administrator	Maintains and updates	Source Code	Each administrator maintains and updates one or more source codes
Administrator	Maintains and updates	Tutorial	Each administrator maintains and updates one or more tutorials
Administrator	Maintains and updates	Game	Each administrator maintains and updates one or more games
Administrator	Maintains and updates	Note	Each administrator maintains and updates one or more notes
Note	Accessed by	User	Each note is accessed by zero or more users
Note	Maintained and updated by	Administrator	Each note is maintained and updated by one or more administrators
Tutorial	Accessed by	User	Each tutorial is accessed by zero or more users
Tutorial	Maintained	Administrator	Each tutorial is maintained and updated by

	and updated by		one or more administrators
Game	Accessed by	User	Each game is accessed by zero or more users
Game	Maintained and updated by	Administrator	Each game is maintained and updated by one or more administrators
Source Code	Accessed by	User	Each source code is accessed by zero or more users
Source Code	Maintained and updated by	Administrator	Each source code is maintained and updated by one or more administrators

5.5 PROCESS VIEW OF THE PROPOSED SYSTEM

5.5.1 Sequence Diagrams

Sequence Diagram is an interaction diagram that emphasizes on time ordering of messages. In drawing a sequence diagram or interaction diagram, each object must have its corresponding classes in the class diagram.

5.5.1.1 "Notes" Package Sequence Diagram

5.5.1.1.1 View Notes

Figure 5.11 illustrates the sequence diagram of "View Notes" use case. The typical course of event can be found in Section 5.3.1.1.

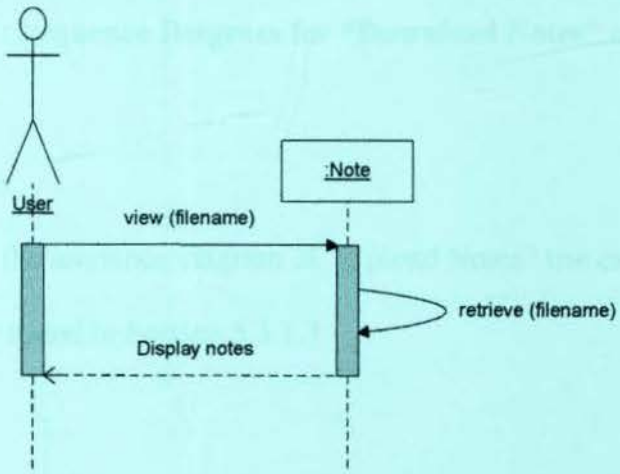


Figure 5.11: Sequence Diagram for "View Notes" Use Case

5.5.1.1.2 Download Notes

Figure 5.12 illustrates the sequence diagram of "Download Notes" use case. The typical course of event can be found in Section 5.3.1.2.

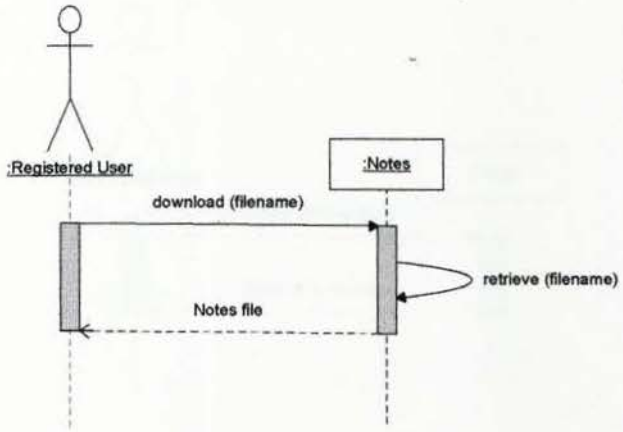


Table 5.12: Sequence Diagram for "Download Notes" use case

5.5.1.1.3 Upload Notes

Figure 5.13 illustrates the sequence diagram of "Upload Notes" use case. The typical course of event can be found in Section 5.3.1.3

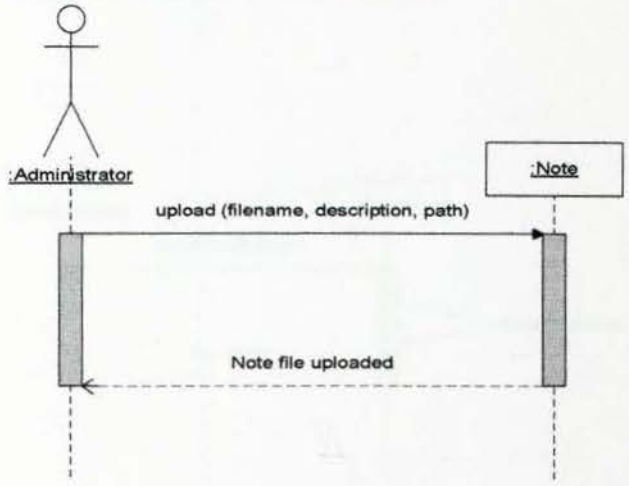


Figure 5.13: Sequence Diagram for "Upload Notes" Use Case

5.5.1.1.4 Delete Notes

Figure 5.14 illustrates the sequence diagram of "Delete Notes" use case. The typical course of event can be found in Section 5.3.1.3

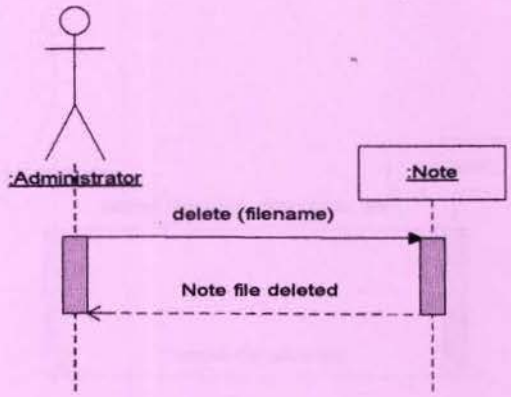


Figure 5.14: Sequence Diagram for "Delete Notes" Use Case

5.5.1.2 "Tutorials" Package Sequence Diagram

5.5.1.2.1 Download Tutorials

Figure 5.15 illustrates the sequence diagram of "Download Tutorials" use case. The typical course of event can be found in Section 5.3.2.1

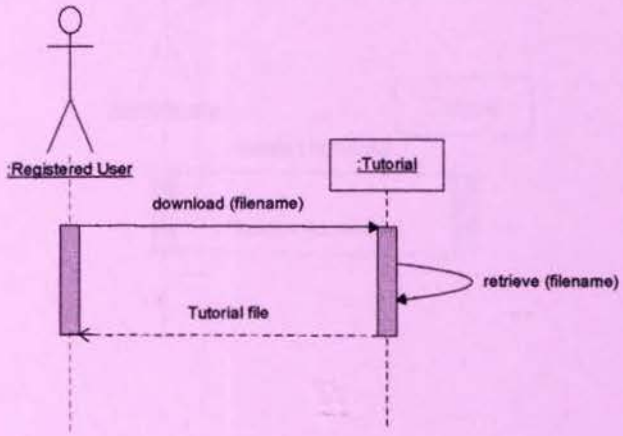


Figure 5.15: Sequence Diagram for "Download Tutorials" Use Case

5.5.1.2.2 Upload Tutorials

Figure 5.16 illustrates the sequence diagram of "Upload Tutorials" use case. The typical course of event can be found in Section 5.3.2.2.

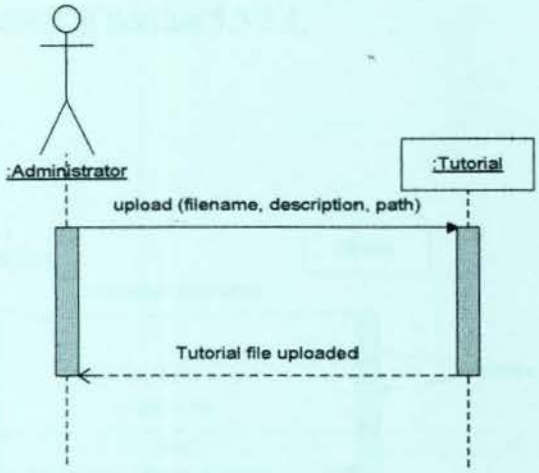


Figure 5.16: Sequence Diagram for "Upload Tutorials" Use Case

5.5.1.2.3 Delete Tutorials

Figure 5.17 illustrates the sequence diagram of "Delete Tutorials" use case. The typical course of event can be found in Section 5.3.2.3.

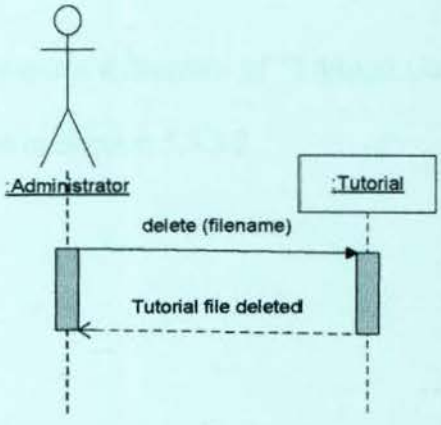


Figure 5.17: Sequence Diagram for "Delete Tutorials" Use Case

5.5.1.3 "Games" Package Sequence Diagram

5.5.1.3.1 Play Games

Figure 5.18 illustrates the sequence diagram of "Play Games" use case. The typical course of event can be found in Section 5.3.3.1.

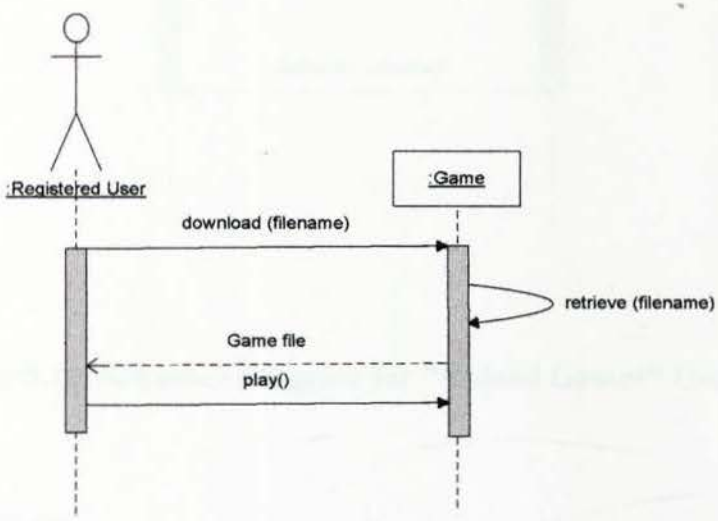


Figure 5.18: Sequence Diagram for "Play Games" Use Case

5.5.1.3.2 Upload Games

Figure 5.19 illustrates the sequence diagram of "Upload Games" use case. The typical course of event can be found in Section 5.3.3.2.

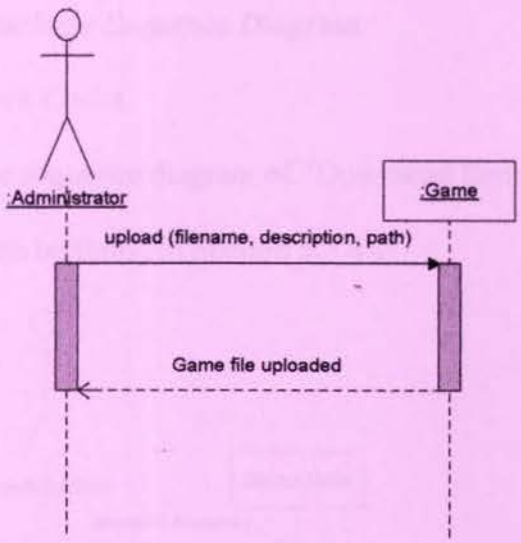


Figure 5.19: Sequence Diagram for "Upload Games" Use Case

5.5.1.3.3 Delete Games

Figure 5.20 illustrates the sequence diagram of "Delete Games" use case. The typical course of event can be found in Section 5.3.3.3.

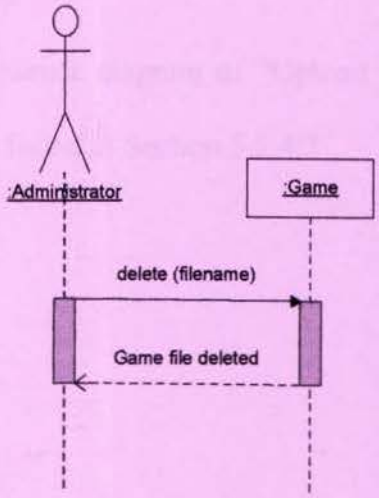


Figure 5.20: Sequence Diagram for "Delete Games" Use Case

5.5.1.4 "Code Dump" Package Sequence Diagram

5.5.1.4.1 Download Source Codes

Figure 5.21 illustrates the sequence diagram of "Download Source Codes" use case. The typical course of event can be found in Section 5.3.4.1.

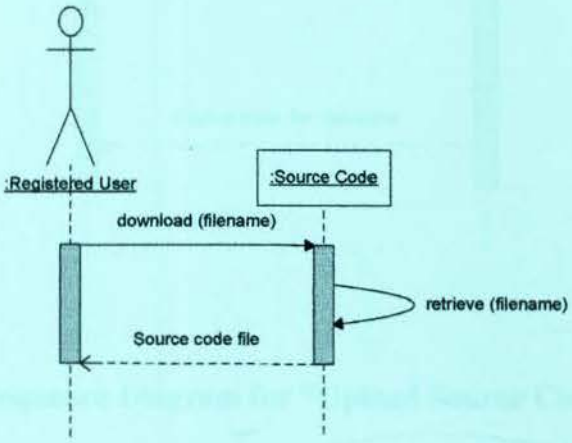


Figure 5.21: Sequence Diagram for "Download Source Codes" Use Case

5.5.1.4.2 Upload Source Codes

Figure 5.22 illustrates the sequence diagram of "Upload Source Codes" use case. The typical course of event can be found in Section 5.3.4.2.

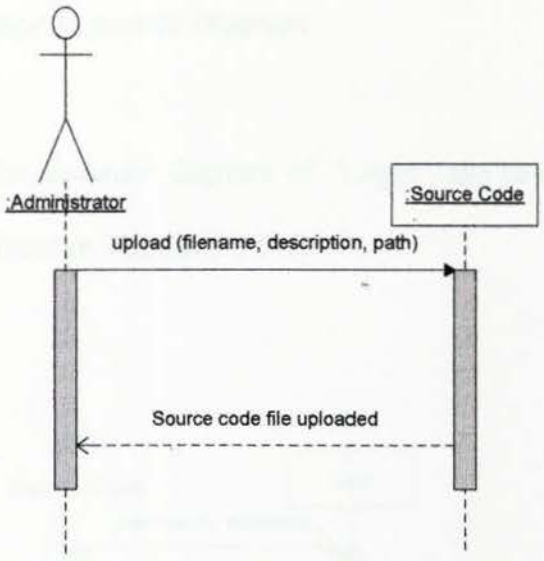


Figure 5.22: Sequence Diagram for "Upload Source Codes" Use Case

5.5.1.4.3 Delete Source Codes

Figure 5.23 illustrates the sequence diagram of "Delete Source Codes" use case. The typical course of event can be found in Section 5.3.4.3.

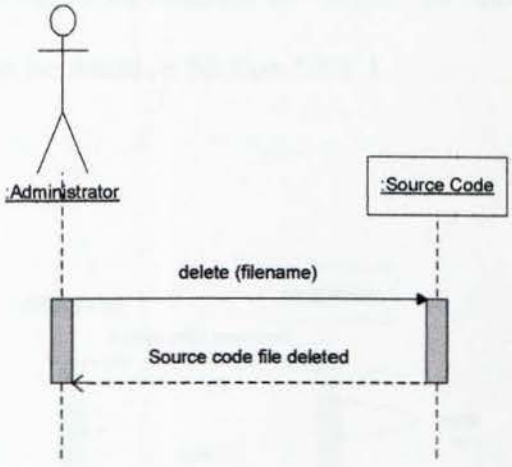


Figure 5.23: Sequence Diagram for "Delete Source Codes" Use Case

5.5.1.5 "Sign-In" Package Sequence Diagram

5.5.1.5.1 Login

Figure 5.24 illustrates the sequence diagram of "Login" use case for User. The typical course of event can be found in Section 5.3.5.1.

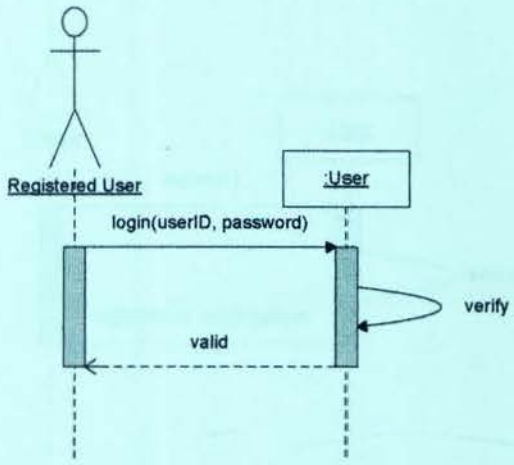


Figure 5.24: Sequence Diagram for "Login" Use Case for User

Figure 5.25 illustrates the sequence diagram of "Login" use case for administrator. The typical course of event can be found in Section 5.3.5.1.

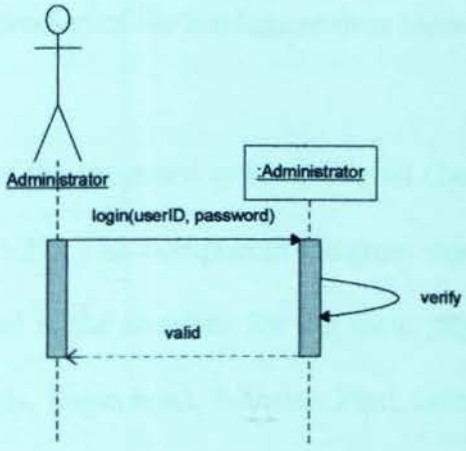


Figure 5.25: Sequence Diagram for "Login" Use Case for Administrator

5.5.1.5.2 Register

Figure 5.26 illustrates the sequence diagram of "Register" use case. The typical course of event can be found in Section 5.3.5.2.

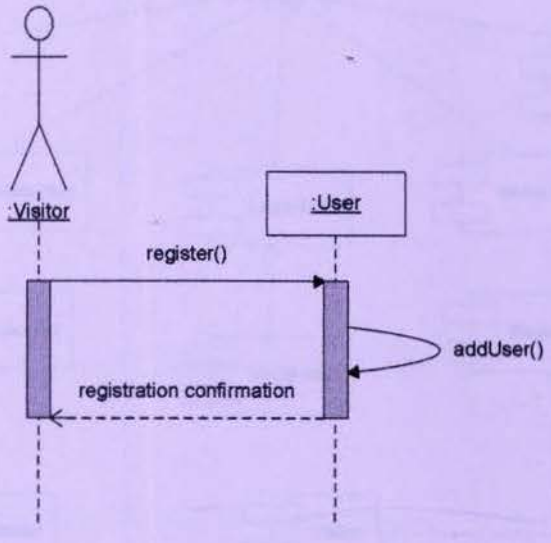


Figure 5.26: Sequence Diagram for "Register" Use Case

5.6 IMPLEMENTATION VIEW OF THE PROPOSED SYSTEM

Implementation view of the system encompasses structural and behavioural dimensions of the system. It focuses on the components and files used to assemble and release the physical system. The end-product of the implementation view is the component diagram.

The component diagram of the proposed system, Digital Game-based C++ Learning is illustrated in the Figure 5.27. The component diagram consists of interfaces of the system. The MainPage.html is the interface for the main page of the system. It can be linked to other pages namely, Login.html, Tutorials.html, Games.html, Notes.html, Code dump.html, Maintenance.html and RegistrationUser.html, verify.asp and addUser.asp.

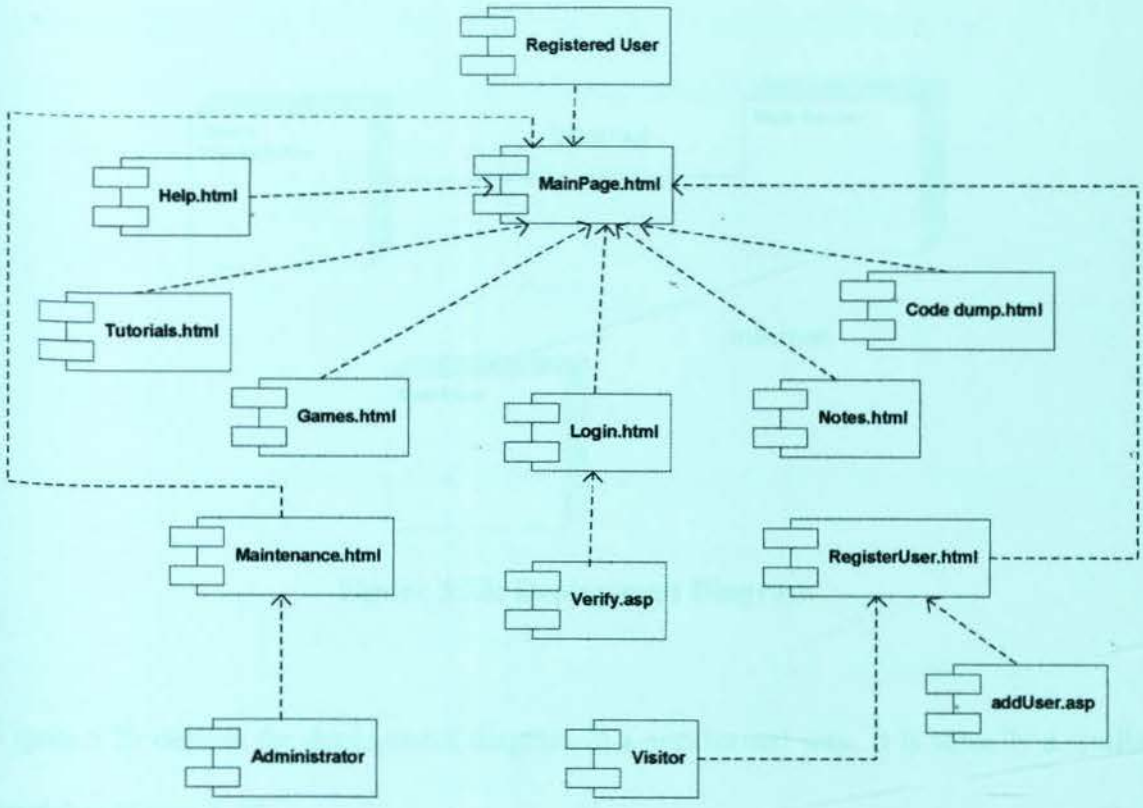


Figure 5.27: Component Diagram

5.7 DEPLOYMENT VIEW OF THE PROPOSED SYSTEM

Deployment View focuses on the geographic distribution of the various software elements on hardware and other physical elements that forms the system. It consists of nodes that forms the hardware topology, addresses the distribution, delivery and installation of the parts that make up the physical system. End product of the deployment view is none other than the deployment diagram.

Figure 5.28 depicts the deployment diagram for the proposed system. It consists of three nodes, namely Client workstation, Web Server and Database. Each of it is linked through the Internet or World Wide Web.

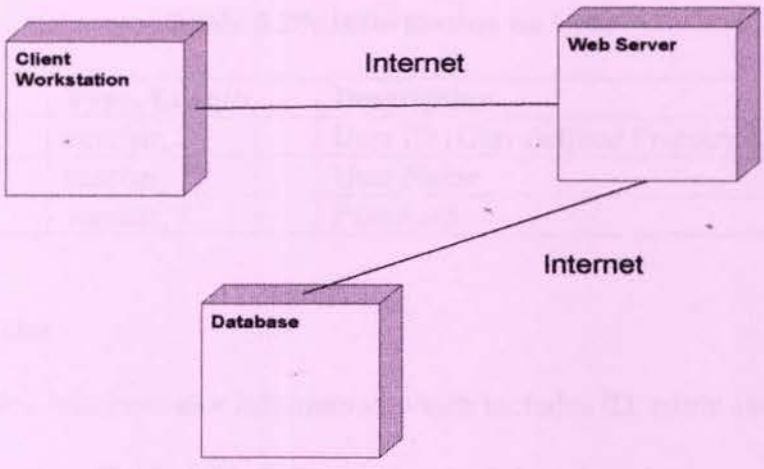


Figure 5.28: Deployment Diagram

Figure 5.29 depicts the deployment diagram in a non-formal way. It is actually a similar model to Figure 5.28.

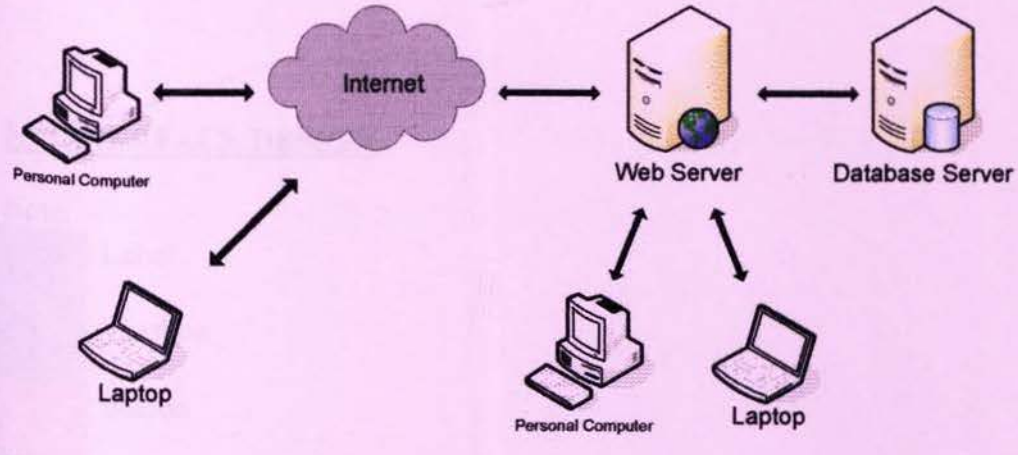


Figure 5.29: Non-formal Deployment Diagram

5.8 DATA DICTIONARY

The Digital Game-based C++ Learning System consists of 6 tables:

1. User

Table 5.20 stores User information which includes ID, name and password.

Table 5.20: Information on User

Field	Type, Length	Description
userID	varchar, 20	User ID (User defined Primary Key)
userName	varchar, 20	User Name
userPwd	varchar, 8	Password

2. Administrator

Table 5.21 stores Administrator information which includes ID, name and password.

Table 5.21: Information on Administrator

Field	Type, Length	Description
adminID	varchar, 9	Administrator ID (User defined Primary Key)
adminName	varchar, 20	Administrator name
adminPwd	varchar, 8	Password

5.9 INTERFACE DESIGN

Note:

Label

Textbox

Button

Title Bar

Drop Down List

Check Box

Pop up windows (by double clicking)

Hyperlink

- Subsystems are Notes, Code Dump, Tutorials, Games, Help Menu and User Profile Maintenance and etc.
- Modules are function(s) that a subsystem can perform
- Main - Redirect to Main.html

  Notes - Redirect to Notes.html

  Tutorials - Redirect to Tutorials.html

  Games - Redirect to Games.html

  Code Dump - Redirect to CodeDump.html

  Help - Redirect to Help.html

  Login - Redirect to LoginPage.html

  Logout - Sign out user and redirect to LoginPage.html

  Maintenance - Redirect to Maintenance.html

(1) MainPage.html

The Main page of the Digital Game-based C++ Learning System is shown below in Figure 5.30.

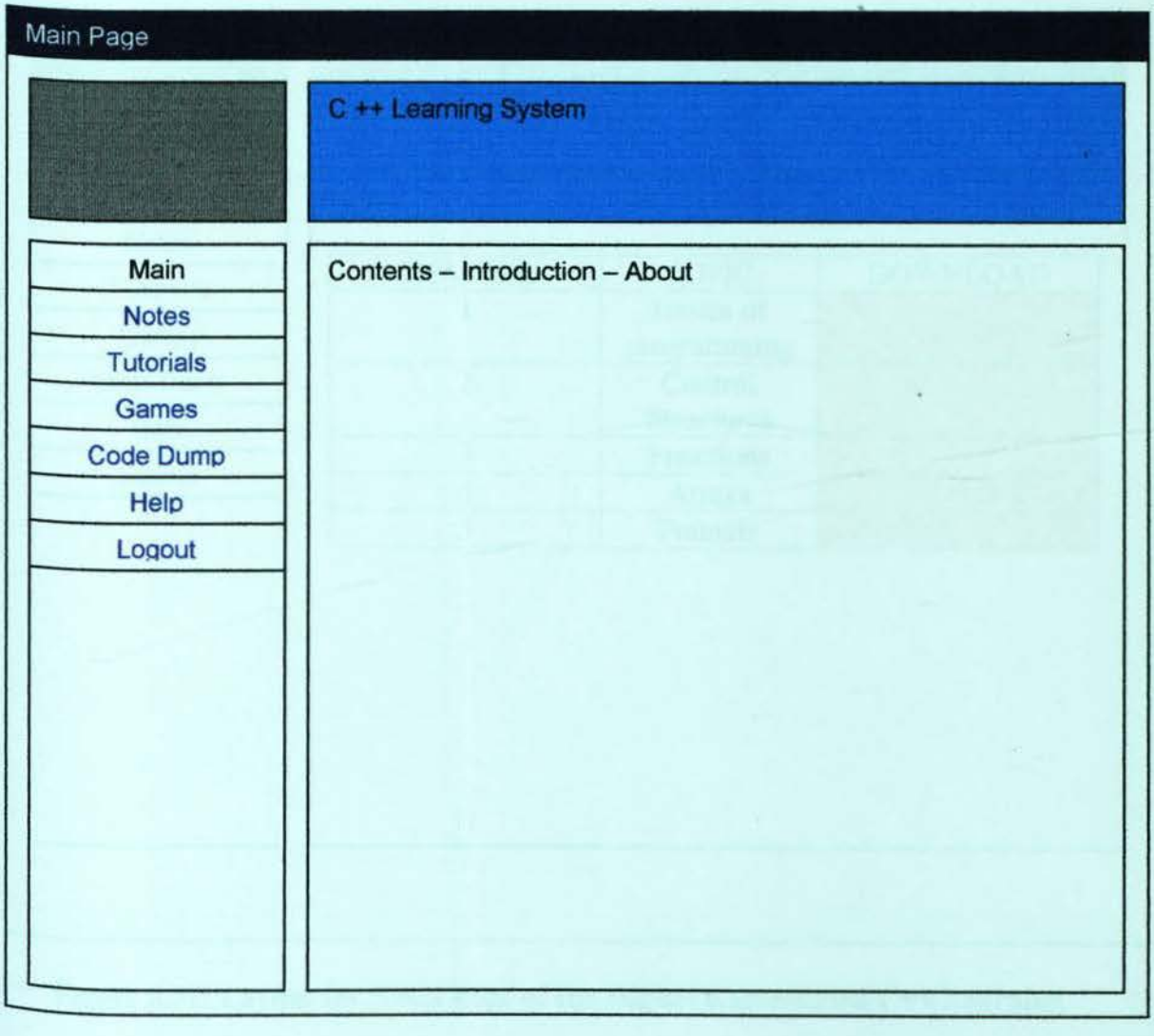


Figure 5.30: Layout for Main page of the Digital Game-based C++ Learning System

(2) Notes.html

The Notes Page of the Digital Game-based C++ Learning System is shown below in Figure 5.31.

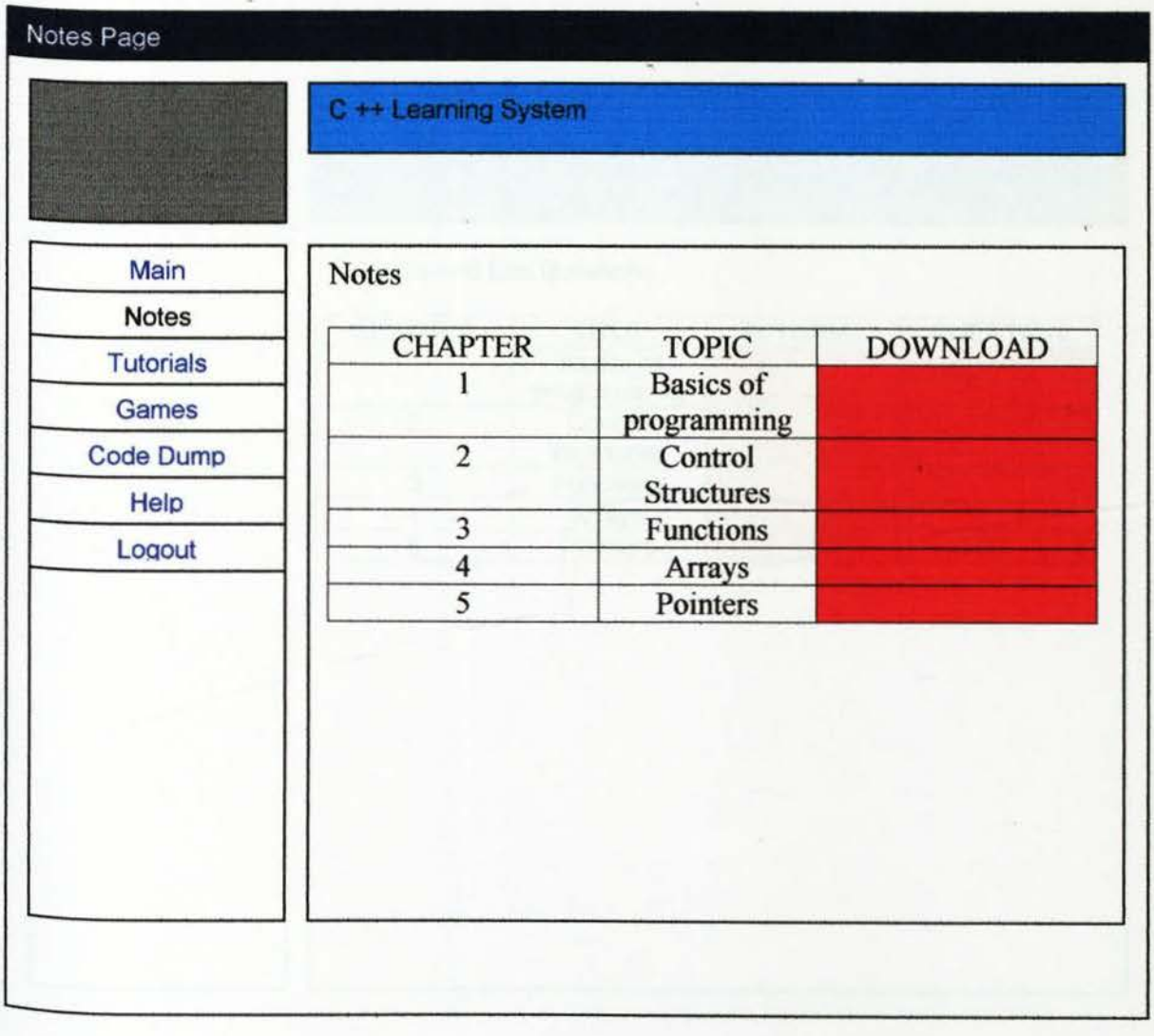


Figure 5.31: Layout for Notes Page of the Digital Game-based C++ Learning System

(3) Tutorials.html

The Tutorials Page of the Digital Game-based C++ Learning System is shown below in Figure 5.32.

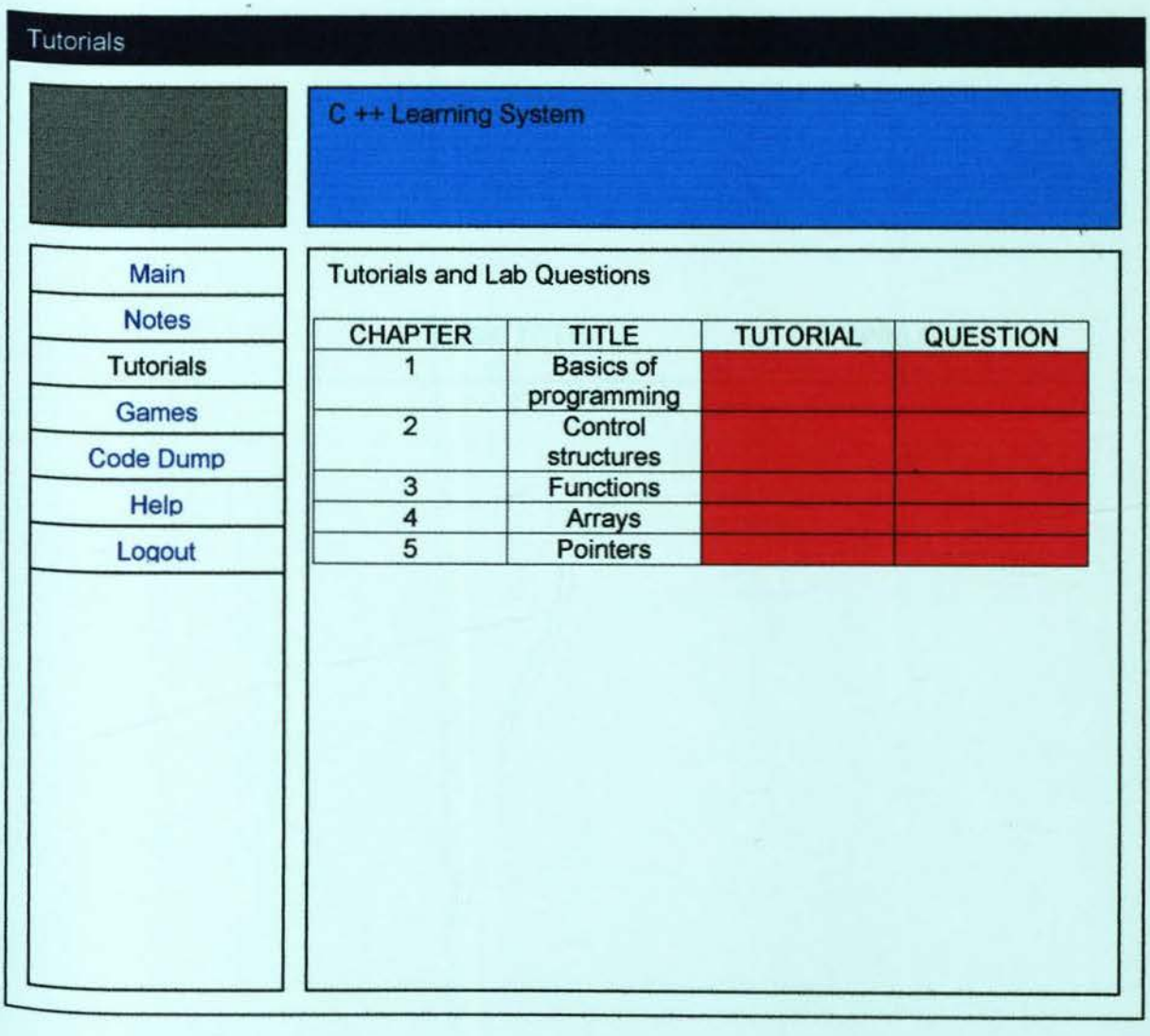


Figure 5.32: Layout for Tutorials Page of the Digital Game-based C++ Learning System

(4) Games.html

The Games page of the Digital Game-based C++ Learning System is shown below in Figure 5.33.

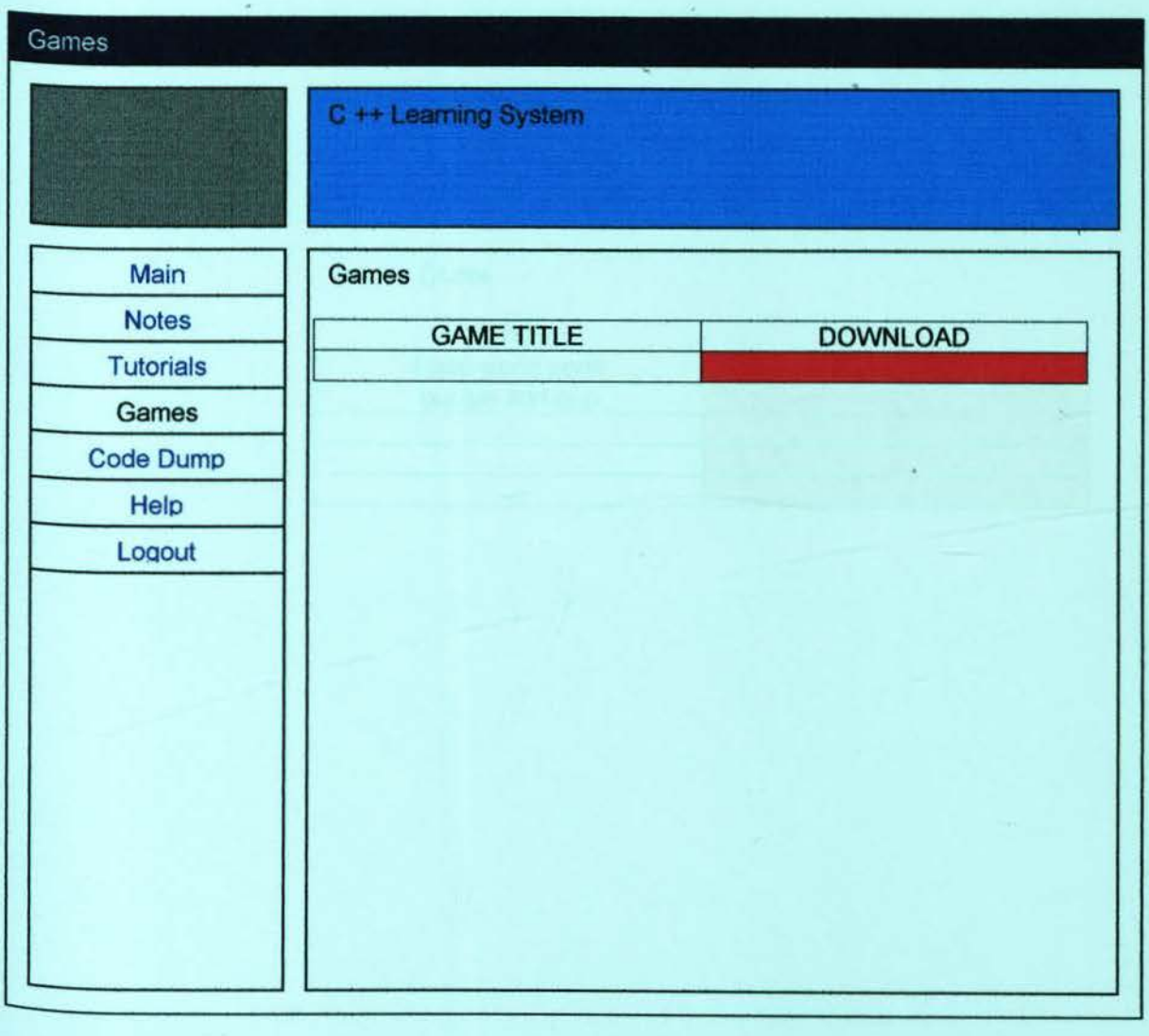


Figure 5.33: Layout for Games Page of the Digital Game-based C++ Learning System

(5) Code dump.html

The Code dump page of the Digital Game-based C++ Learning System is shown below in Figure 5.34.

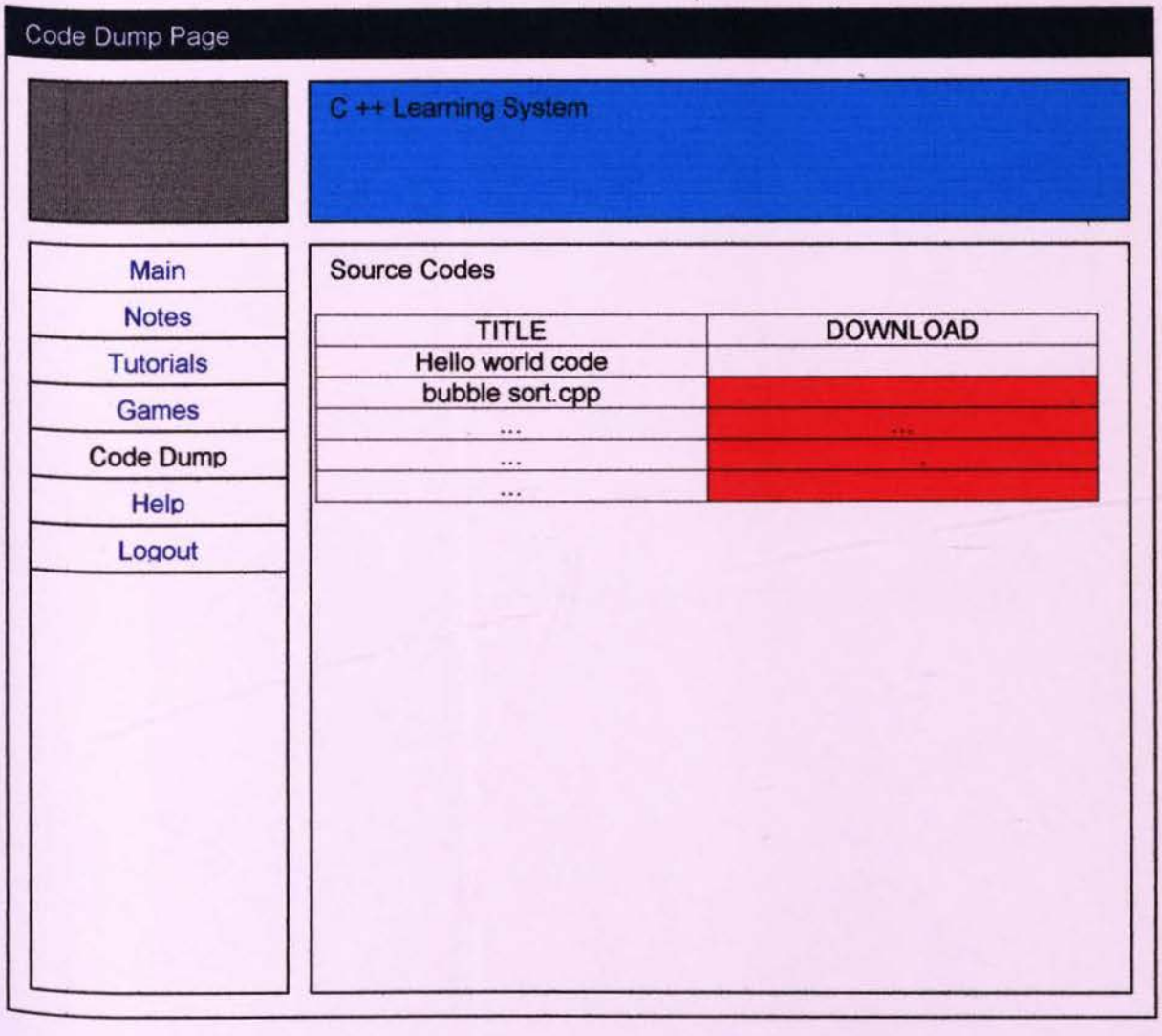


Figure 5.34: Layout for Code Dump Page of the Digital Game-based C++ Learning System

(6) Help.html

The Help Page of the Digital Game-based C++ Learning System is shown below in Figure 5.35.

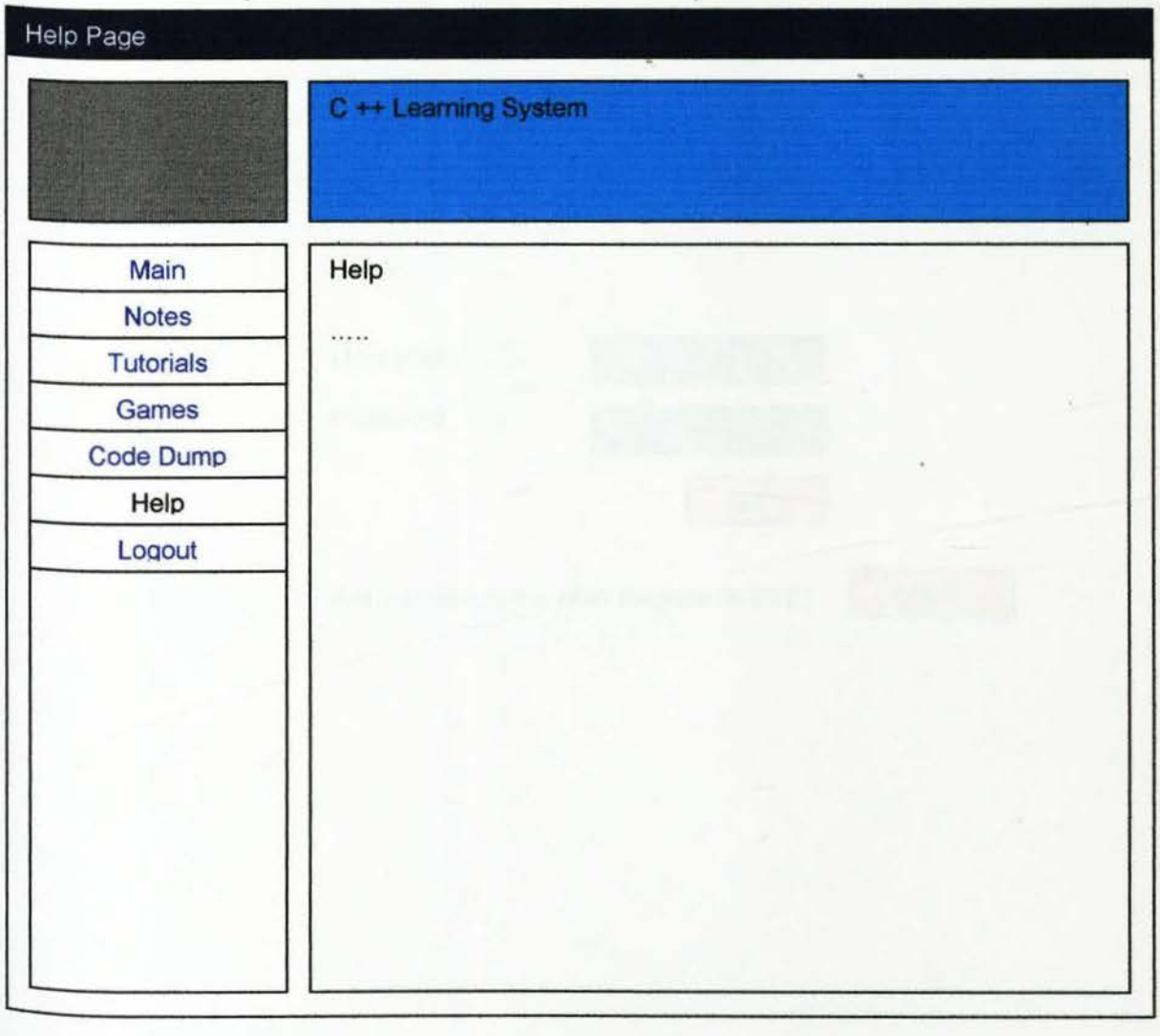


Figure 5.35: Layout for Help Page of the Digital Game-based C++ Learning System

(7) LoginPage.html

The LoginPage of the Digital Game-based C++ Learning System is shown below in Figure 5.36.

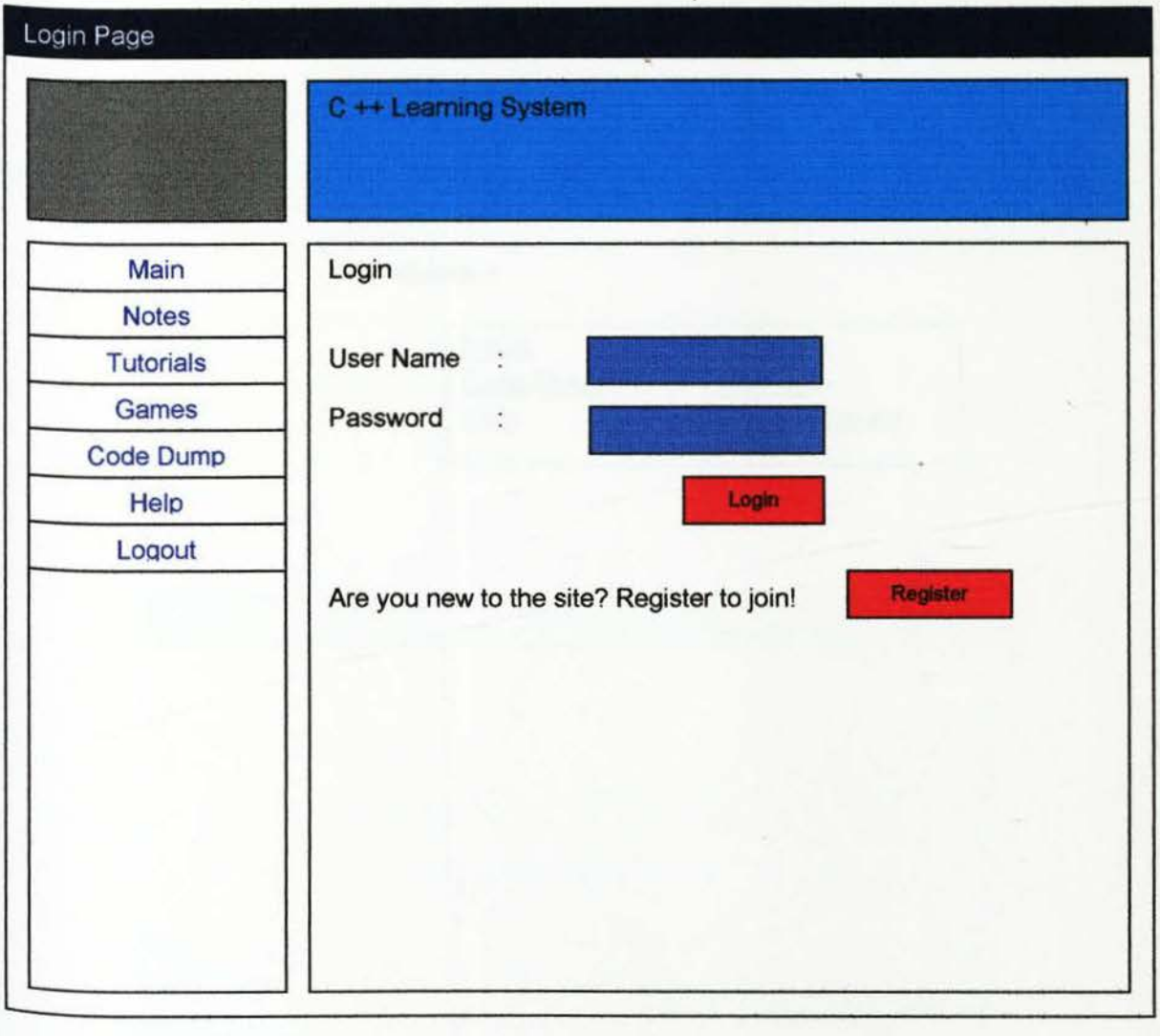


Figure 5.36: Layout for LoginPage of the Digital Game-based C++ Learning System

(8) Maintenance.html

The Maintenance Page of the Digital Game-based C++ Learning System is shown below in Figure 5.37.

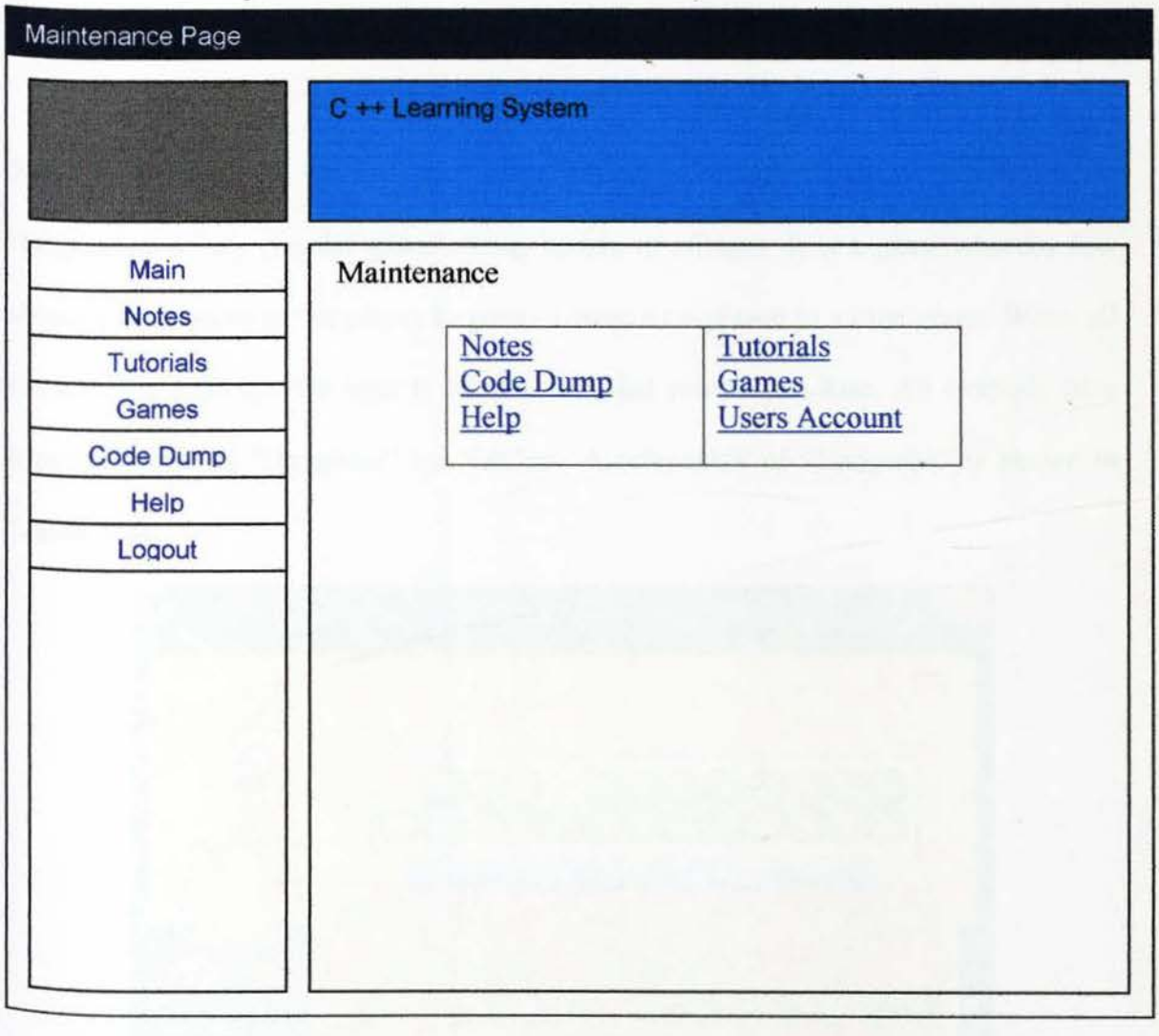


Figure 5.37: Layout for Maintenance Page of the Digital Game-based C++ Learning System

5.10 GAMES DESIGN

This proposed system has roughly few types of games:

- Hangman
- Puzzle Games

5.10.1 Hangman

Hangman is a very popular game among people of all ages. It is a game whereby few chances were given to the player to guess a word or a phrase to a clue given. When all chances are used up, the man is hanged and that means you lose. An example of a hangman game is "Hangaroo" by NetCent. A screenshot of "Hangaroo" is shown in Figure 5.38.

Figure 5.38: Screenshot of "Hangaroo"

Figure 5.39 shows the workflow of the hangman game my proposed system. This game is only suitable for theoretical part of C++ programming.

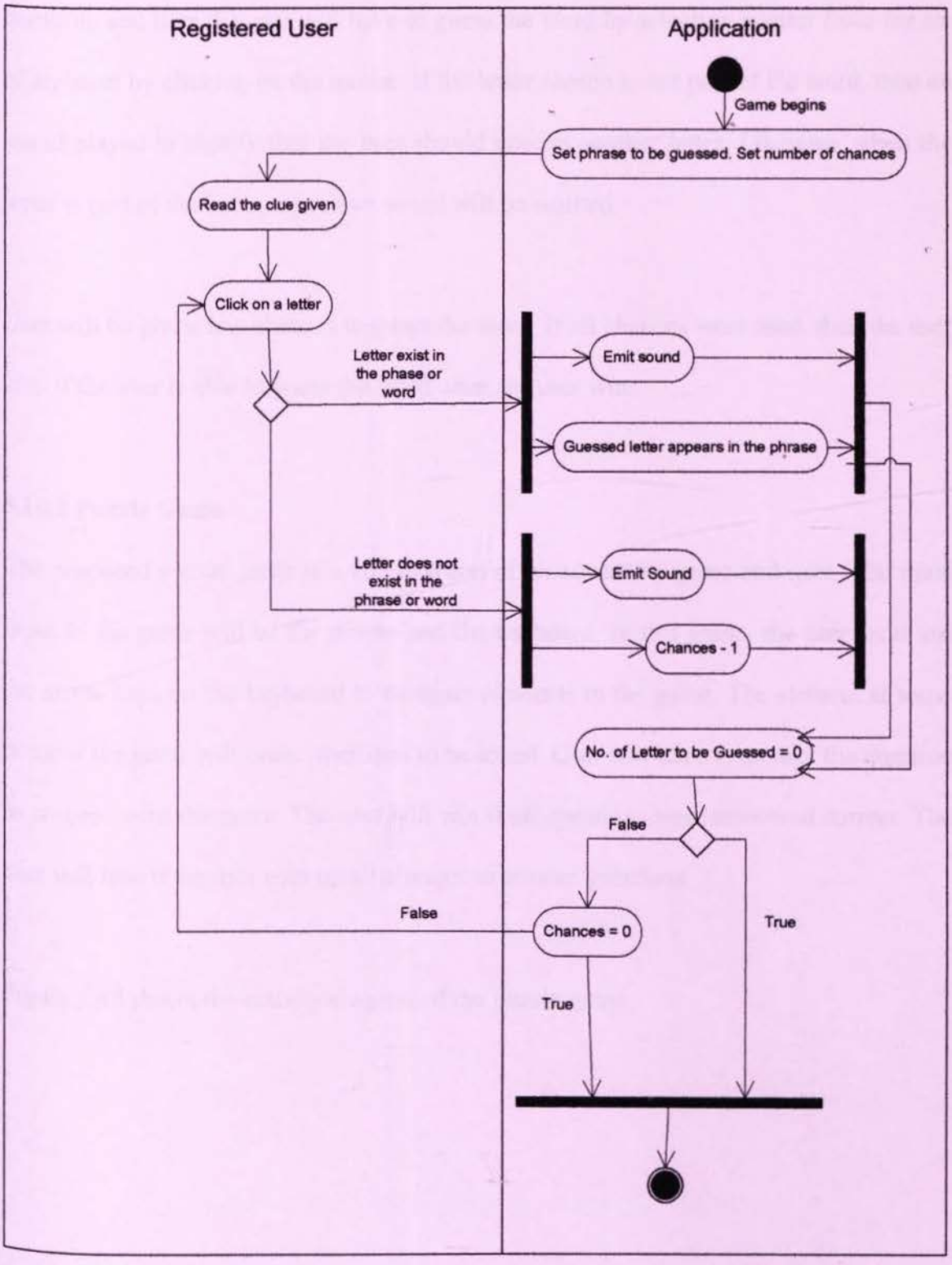


Figure 5.39: Activity diagram showing the workflow the proposed Hangman game

This game uses the mouse as the main input. The proposed game has a set of alphabet ranging from A to Z. When the game is started, the clue will be given in the form of question, and then the user will have to guess the word by selecting a letter from the set of alphabet by clicking on the mouse. If the letter chosen is not part of the word, then an sound played to signify that the user should choose another letter. Likewise, when the letter is part of the letter, a positive sound will be emitted.

User will be given few chances to guess the word. If all chances were used, then the user lost. If the user is able to guess the word, then the user wins.

5.10.2 Puzzle Game

The proposed puzzle game is a combination of an adventure game and quiz. The main input to the game will be the mouse and the keyboard. In this game, the user must use the arrow keys on the keyboard to navigate elements in the game. The element at some point of the game will cause questions to be asked. User will have to answer the question to proceed with the game. The user will win if all questions were answered correct. The user will lose if the user uses up all chances to answer questions.

Figure 5.40 shows the activity diagram of the puzzle game

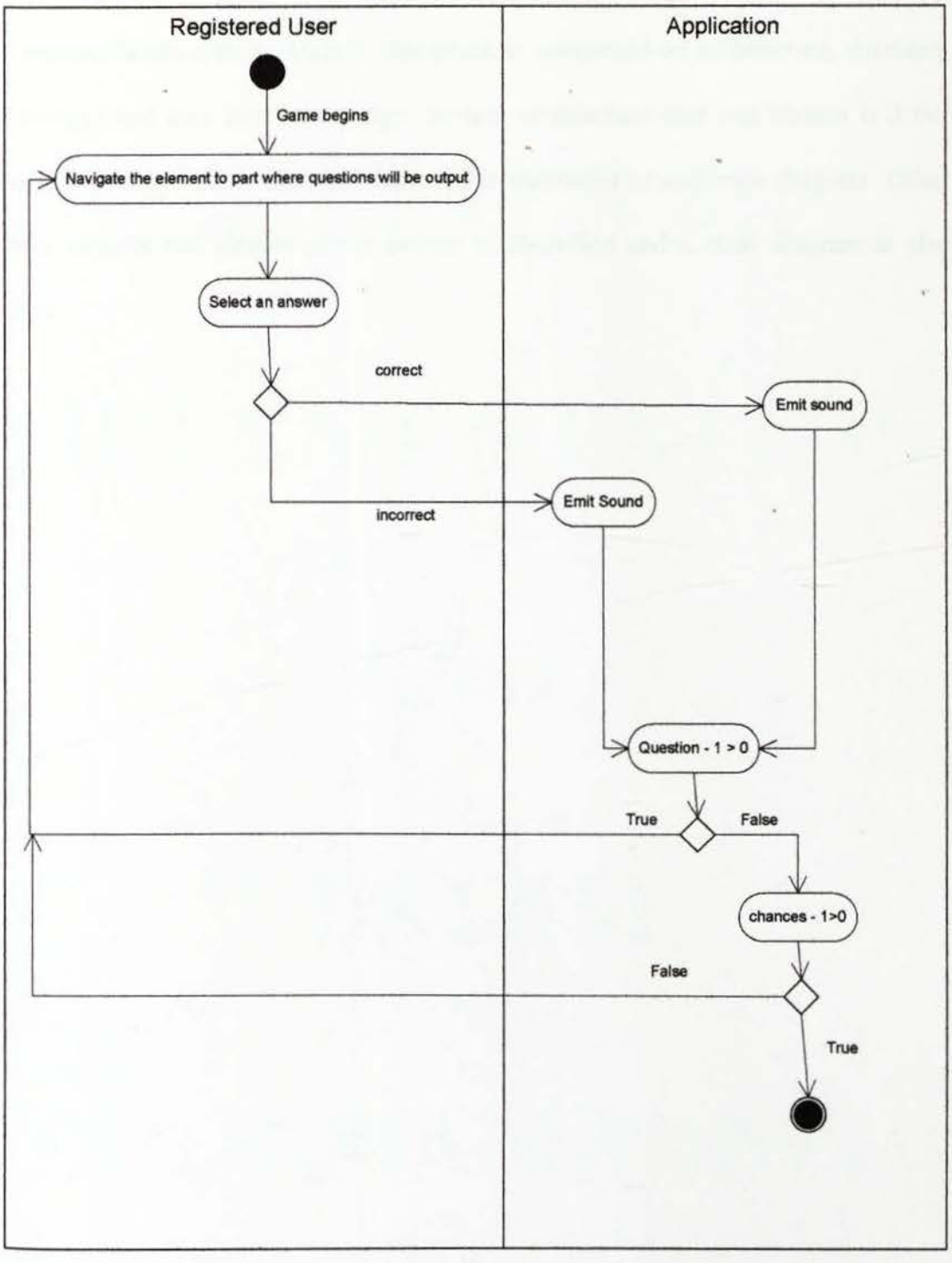


Figure 5.40 Activity Diagram of the proposed Puzzle game

5.11 SUMMARY

In system design phase, all requirements produced in the analysis phase are translated into a representation system. Mainly, this phase is concerned on architecture, database, game designs and user interface design. System architecture that was chosen is 3-tier architecture. All use cases that was captured is translated to sequence diagram. Other than that, objects and classes of the system is identified and a class diagram is also produced.

CHAPTER 6:

IMPLEMENTATION

AND

SYSTEM

PROGRAMMING

CHAPTER 6: IMPLEMENTATION AND SYSTEM PROGRAMMING

6.1 INTRODUCTION

Once design features are defined, the implementation started. There are two things to focus on: (1) games; and (2) website. For games, the implementation phase started with the multimedia authoring tools while the website started with web development tools.

6.2 GAME DEVELOPMENT

Before development can begin, it is important to map out the theme of the games to be built. Flow of the games are then, mapped. Then, prepare graphics and images and sound files which are needed for the particular. In the following subsection will look at the tools used, and steps taken to some of the important part of both games.

6.2.1 Game Development Tools

Basically, what is used to develop the game were:

(1)	Operating System	:	Windows XP Professional
(2)	Multimedia Authoring Tool	:	Macromedia Flash MX 2004
(3)	Programming language	:	ActionScript 2.0
(4)	Graphic editor	:	Adobe Photoshop Elements
(5)	Sound editor	:	Audacity

Before developing the game, all development tools must be installed in the computer.

6.2.2 Development Process

6.2.2.1 Preparing the Graphics or Images

For both games, most of images were hand drawn, scanned into the computer and from there, they were edited and manipulated using Adobe Photoshop Elements. Adobe Photoshop Elements provides exceptionally good filtering features that transformed the roughly drawn images into cartoon-like images. The images were all in black-and-white, so, Adobe Photoshop Elements is also used to colour all the images. All these manipulated images were then saved in .psd and .png formats. (.psd format is the Adobe Photoshop format, .png format is the format Macromedia Fireworks format). .psd format is important to preserve the images for further manipulation while .png format is provides hassle-free import to Macromedia Flash.

6.2.2.2 Game Development with Macromedia Flash MX 2004

To kick start the development, first thing that must be done with the authoring tool is to set up the flash movie properties. This is where the size of the movie, background colour and etc. of the flash movie to be built are determined. This can be done by accessing the Modify -> Document in the menu bar.

6.2.2.2.1 Importing Artwork

After setting up the movie properties, import all artwork (images) and audio files to be used in the game. This can be done by accessing File -> Import -> Import to Library. Then, choose artwork and audio files needed for the game. The steps are shown in Figure 6.1.

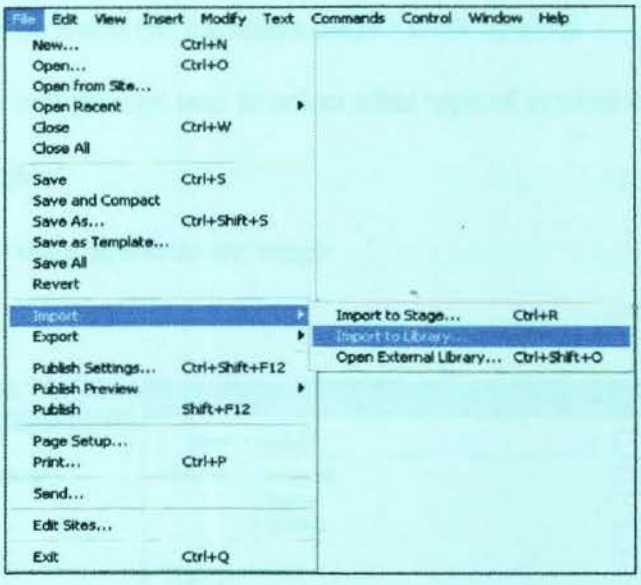


Figure 6.1: How to import artwork to the Library

6.2.2.2.2 Drawing in Flash

Not all images used were hand drawn. There are graphics such as graphics to manipulate buttons, frames for the game and other simple images were created in Flash. Images created in Flash can only be used in Flash itself not anywhere outside the Flash environment.

Flash has a tool box complete with brushes and shape tools that enable drawing. There are 2 ways to create a bitmap image.

First way:

1. Draw the image on the stage.
2. Right-click and select Convert to Symbol... option.
3. Flash will prompt user to select what type of symbol you wish to create. Select "Graphic".

Alternative way (depicted in Figure 6.2):

1. Click on Insert (in the menu bar) -> New Symbol
2. Flash will prompt user to select what type of symbol to create. Select "Graphic"
3. Draw the graphic on the stage.

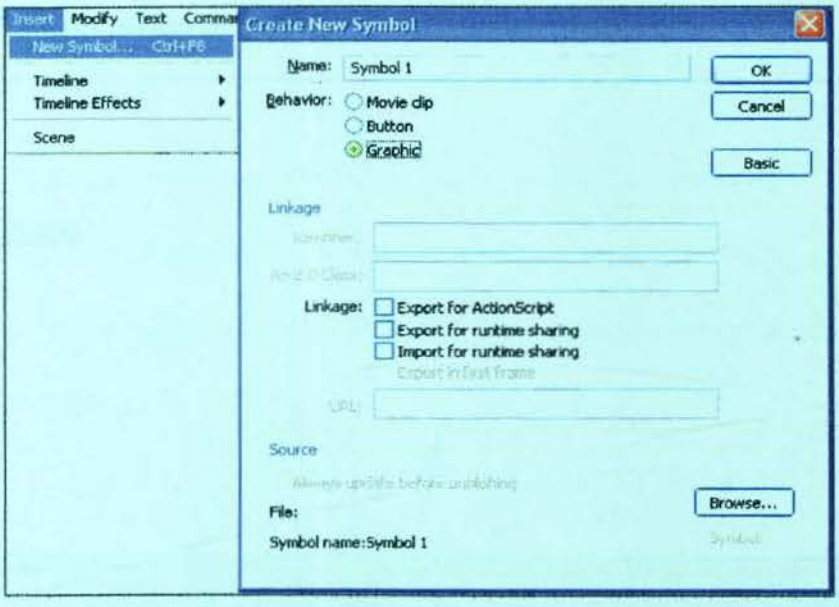


Figure 6.2: Creating a new graphic symbol

6.2.2.2.3 Text

Flash allows user to include text in the game in many ways. To sum it all, Flash provides 3 types of text types, each suited to a specific kind of content:

- **Static text**: This type is to display type or text content created at author-time that won't change at runtime
- **Dynamic text**: This is usually used to hold text content that is generated at runtime from a live data source, or text that will be updated dynamically.
- **Input**: These are fields created for text that is entered at runtime by user.

To include any of the text types stated, just use the Text tool provided by Flash. I used only Static text and Dynamic text in both games but not input.

Static and Dynamic text are contained in text boxes. So, to create any of these text boxes, select the Text tool in the tools box, set the type of text you want in the properties inspector and then, drag the text box on the stage. Figure 6.3 shows the Text tool in the tool box and the properties inspector.

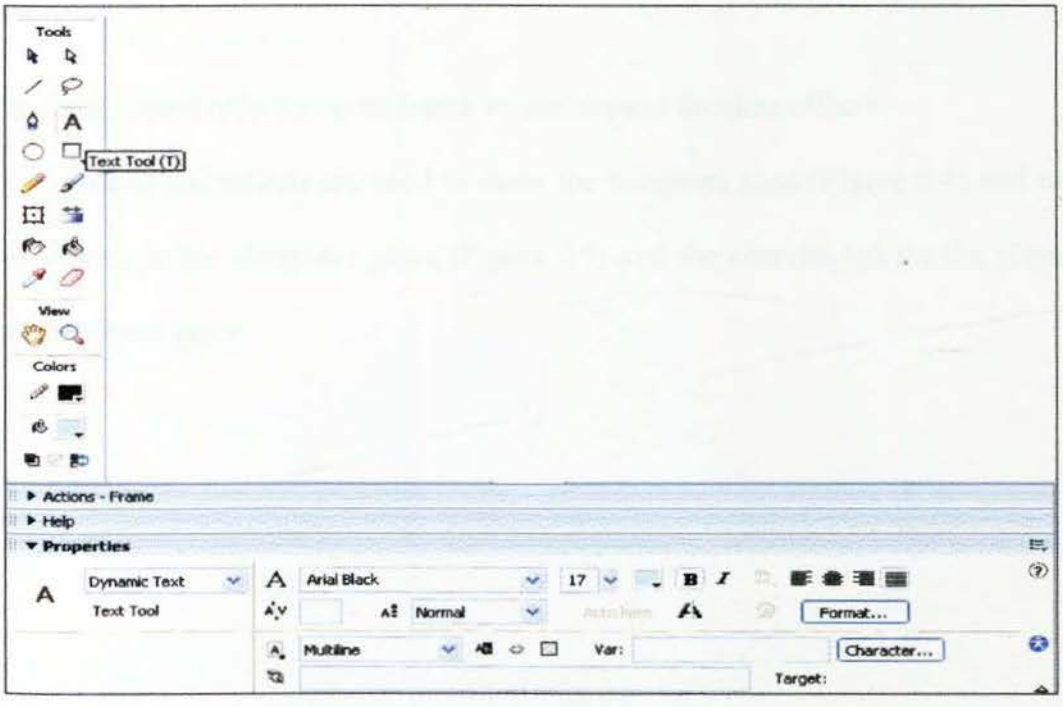


Figure 6.3: The Text Tool and Properties Inspector

6.2.2.2.4 Animation and Movie Clips

Animation is the process of creating the illusion of movement or changes over time. There are 3 basic methods of animation allowed by Flash:

- **Frame-by-frame animation**: Achieved by manually changing the individual contents of each of any number of successive keyframes

- **Tweened animation:** Achieved by defining the contents of the start and end points with keyframes and allow Flash to interpolate the contents of the frames in between.
- **Timeline effects:** Flash MX 2004 new feature gives "automated" animation and visual effects that can be applied to a shaped and symbols. These are prebuilt scripts that you can control by choosing settings in a preview dialog box before the effect is rendered.

Of the three, I used only frame-to-frame animation and timeline effects.

Frame-to-frame animations are used to show the hangman state (Figure 6.4) and round advancements in the Hangman game (Figure 6.5) and the chances left for the player in Computer Spree game.

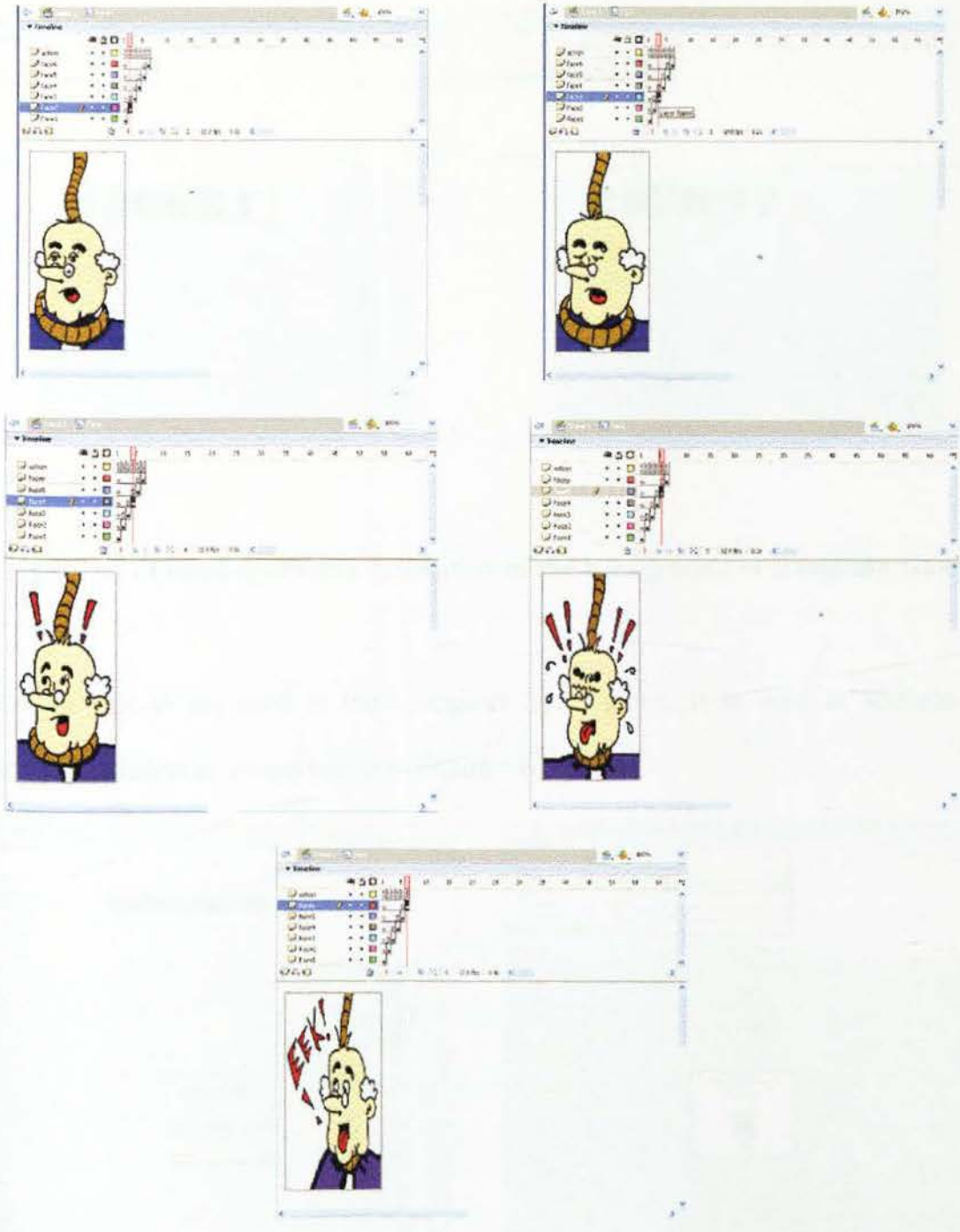


Figure 6.4: Frame-by-Frame Animation of the Hangman state of Hangman Game

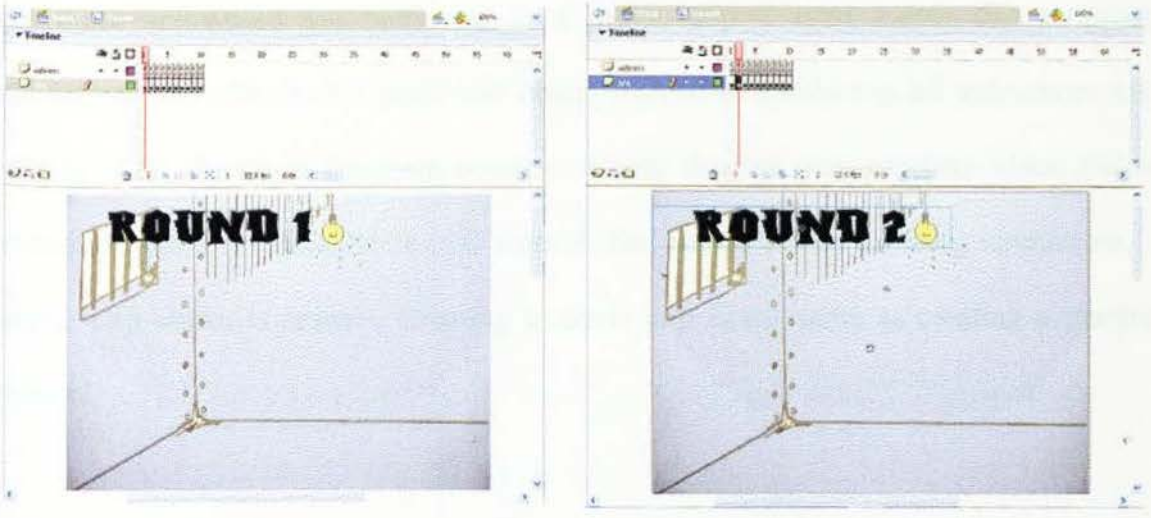


Figure 6.5: Frame-by-Frame Animation of the background of Hangman Game

Timeline effects are used in the Computer Spree game. It is used to animate the shopping counters as shown below in Figure 6.6

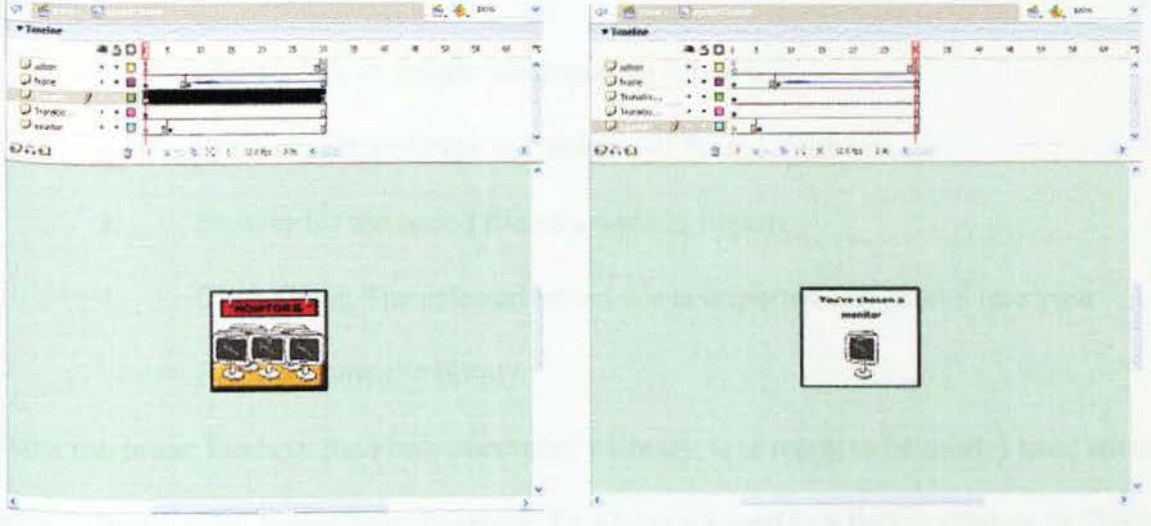


Figure 6.6: Timeline Effects for Computer Spree game

All these animations are contained in a movie clip. I used movie clip to group animations and effects of a particular object instead of combining all animations and effects of all objects in the main timeline. Movie clip has own timeline which makes animation easier to manipulate and control. So, before I start creating animations, a movie clip object is created. Creating a movie clip is the same as creating a graphic symbol.

6.2.2.2.5 Adding Sound

Sound is used in Flash movies to enhance interactive design with navigation elements such as buttons. Flash MX 2004 can work with a wide variety of sound file formats. All the formats I used for both games are WAV (windows wave). To use sounds in the games, sounds have to be imported into the Flash stage or the library. To import, these are the steps:

1. Choose File -> Import -> Import to Library
2. From the Files of type list, select All Sound Formats
3. Browse for the sound file you want to import
4. Click Open. The selected sound file is imported and placed into your Flash document's library.

Now the sound file is in the Flash document's library, it is ready to be used. I used sound files in buttons and throughout the game. To assign a sound to a button (shown in Figure 6.7):

1. Create a Button symbol (Insert Symbol -> New Symbol)
2. Drag an instance of the button from the Library to the Stage
3. Edit the Button symbol by double-clicking it on the Stage

4. Add a new layer to the button's timeline and then add keyframe to this layer in one of the button state (Up, Over, Down, Hit).

5. Select the frame of the button state you want to add a sound and then access the Property inspector by doing one of the following: (a) right-click, choose Properties from the contextual menu; or (b) choose Window->Properties. Another alternative is to simply drag the sound from the Library panel onto the Stage.

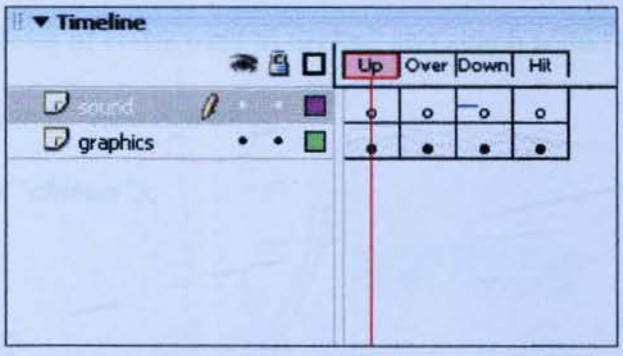


Figure 6.7: Adding sound to a button

Other than using sounds in buttons, sounds are used while the game is being played. Sounds as such were assigned using ActionScript. To use sound files for programming with ActionScript, a linkage identifier need to be created. To create a linkage identifier for a sound from the Library:

1. Select the sound file in the panel and right-click the sound file.
2. Choose the Linkage option in the contextual menu
3. In the Linkage properties dialog box (Figure 6.8), select the Export for ActionScript checkbox. The sound filename automatically populated the Identifier field. You may rename or leave the identifier term as it is.

Figure 6.8: Linkage properties of a sound file

Below are the some lines of codes used to declare and attach sound to a variable.

```
var soundCorrect:Sound = new Sound( );

soundCorrect.attach("chimes");

//play the sound file

soundCorrect.start( );
```

6.2.2.2.6 Creating Buttons

To create a button:

1. Click Insert->New Symbol->Button.
2. Give the button a name and click OK. The button is now stored in the Library and automatically opens in the Edit mode, so you can add some Content to the button
3. You will notice that the button's timeline has 4 keyframe labels that define the button state by mouse behaviour: Up, Over, Down, Hit. These various keyframes can have multiple layers and contain any

Visual element or sound that you want.

Figure 6.9 shows the creation of "Quit" button.

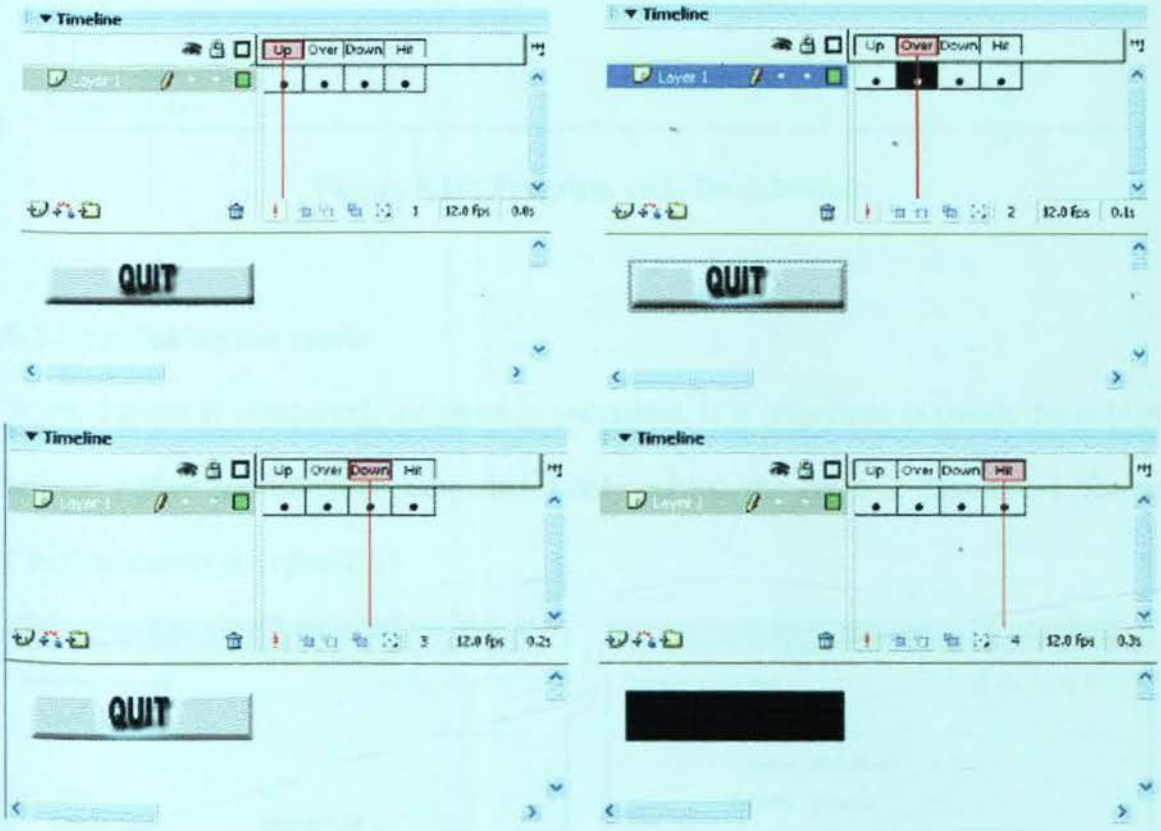


Figure 6.9: Creating a button

6.2.2.2.7 Enabling Buttons

Buttons need programming codes to make it perform a function. Figure 6.10 is a sample of code for "START" button. When clicked, it will start the game by navigating the to the "game" frame in the main timeline.

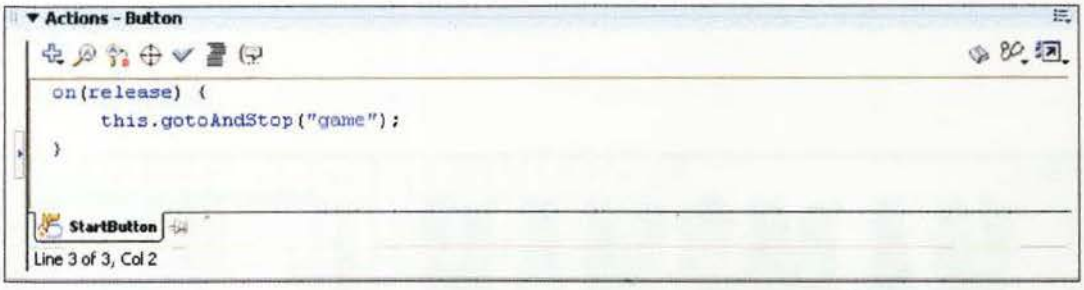


Figure 6.10: Program code for a button

6.2.3 Publishing the movie

When a game is completed, the game is published. It is important to tweak the publish settings before publishing is completed. I only publish the game in flash movie format (.swf) as shown in Figure 6.11.

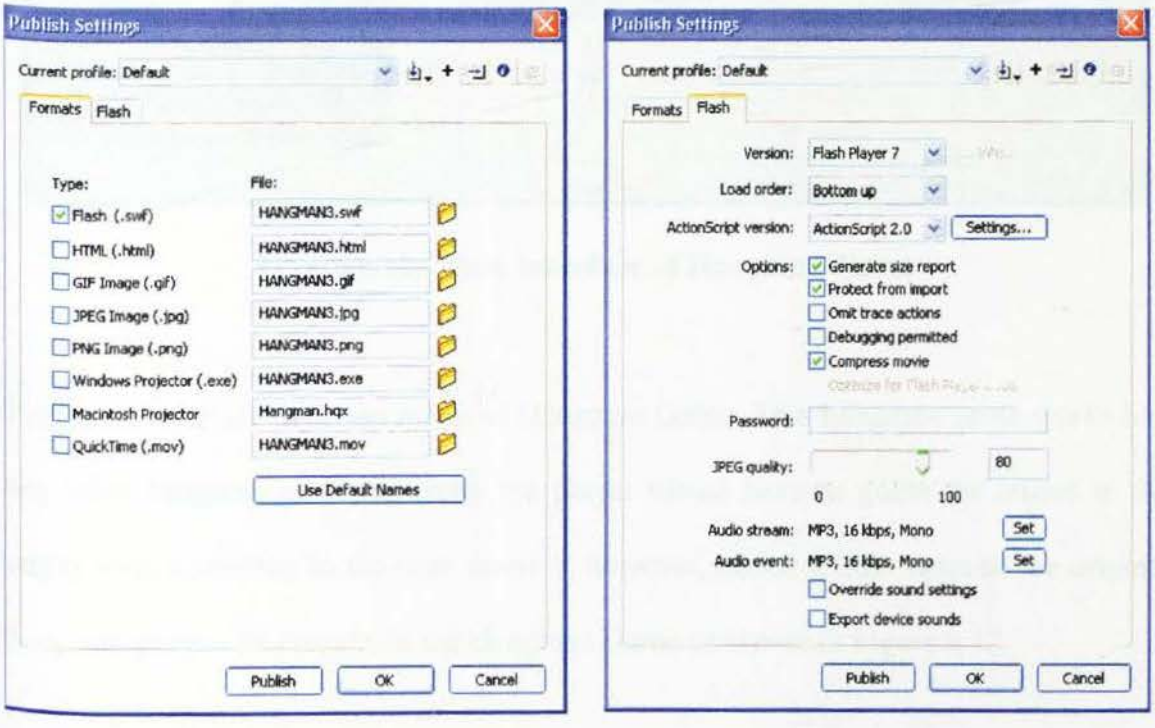


Figure 6.11 Publish settings for the game

6.3 HANGMAN

6.3.1 Introduction

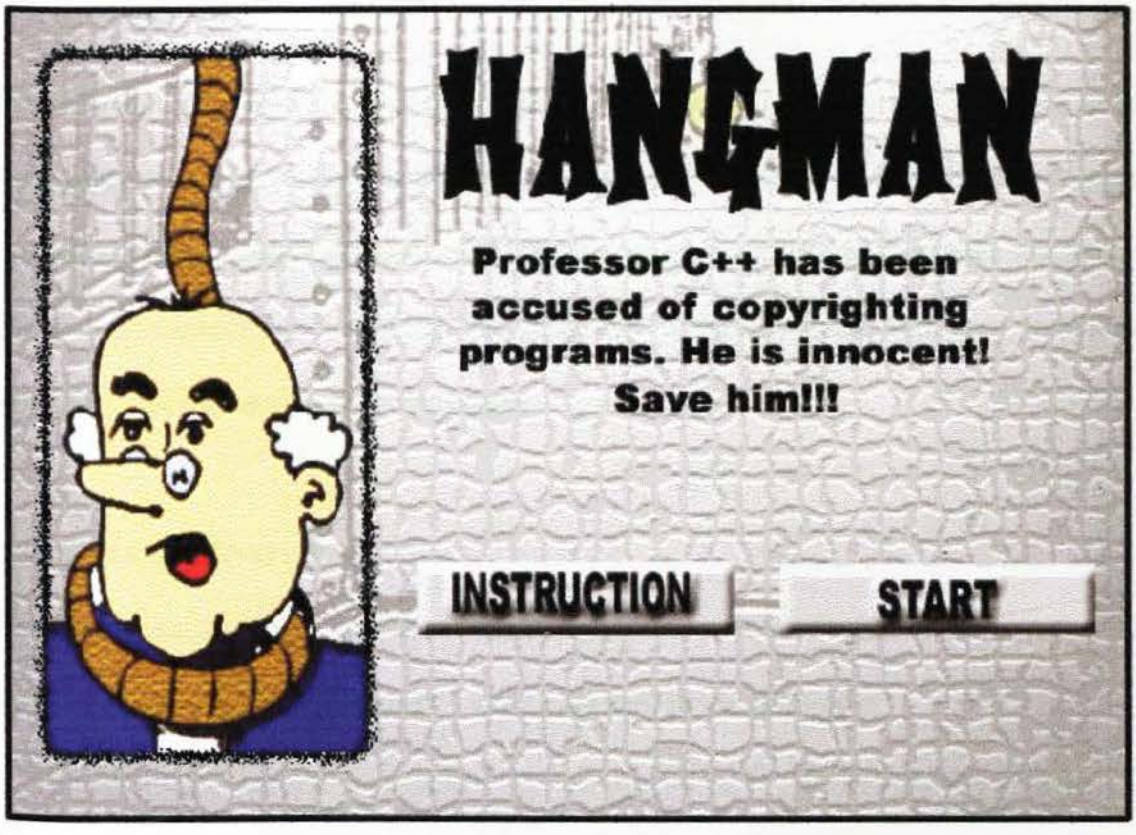


Figure 6.12: Main Interface of Hangman Game

Figure 6.12 depicts the main menu of Hangman Game. This hangman game works like any other hangman game, whereby the player would have to guess the letters in the empty slots according to the clue given. I, however, added a little twist to the original hangman game – this results in the Hangman Game as shown in Figure 6.12.

This game, on the whole, focuses more to the theoretical part of C++ procedural programming. Although theoretical, some of the questions included in the game were

never given second thoughts because programming usually stresses on how to write a program, not little things like definitions which are deemed as secondary information.

Basically this is a classic game that is played by people of all ages. The mouse is the main input to the game. This is to cater for some of the users' who felt that some game should be played using the mouse alone. Little can be gained from this game but, games like this do capture people's attention. They will eventually grasp and learn something when they are interacting with the game.

6.3.2 Hangman Overview

The Hangman Game is about saving a professor, named Professor C++, who was accused of copyrighting programs, from being sent to the gallows.

6.3.3 Flow of the Hangman Game

6.3.3.1 Flowchart of the Hangman Game

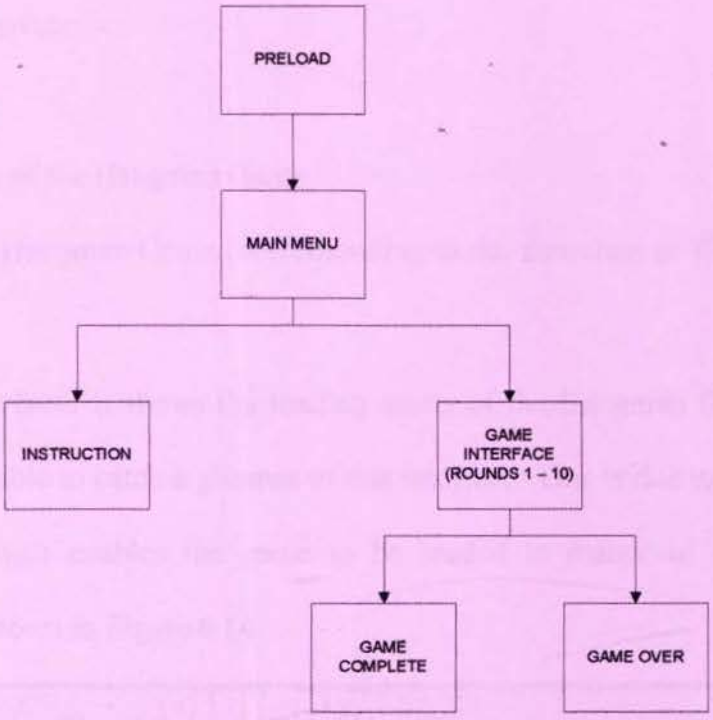


Figure 6.13: Flowchart depicting the flow of interfaces in Hangman Game

Figure 6.13, as shown above, is a flowchart that shows the flow of interfaces in Hangman Game. The flow described as follows:

- Preload menu is the interface that shows the loading assets of the Hangman game. After loading is completed, it will automatically open up the Main Menu of the Hangman game, as shown in Figure 6.12
- From the Main Menu, either the Instruction interface or the Game Interface depending on which button is clicked.

- Game Interface refers to the 10 rounds of letters guessing. There are 2 possibilities in the next interface. The next interface will be either Game Over interface or Game Complete interface.

6.3.3.2 Interfaces of the Hangman Game

Interfaces of the Hangman Game (corresponding to the flowchart in Figure 6.13) are as follows:

- **Preload interface:** It shows the loading assets of the Hangman Game. The player usually only able to catch a glimpse of this interface. This is due to the small size of the game which enables the game to be loaded in matter of seconds. Preload interface is shown in Figure 6.14.

Figure 6.14 Preload Interface of the Hangman Game

- **Main Menu Interface:** Figure 6.12 depicts this interface. It is the "introduction" to the game. It has a description of the objective of the game and two buttons. The functions of the buttons are described in Table 6.1.

Table 6.1 Functions of Buttons in the Main Menu

	When clicked, it will direct the player to the Instruction Interface
	When clicked, it will start the game by directing the player to the first round of the game

- **Instruction Interface**: Figure 6.15 illustrates the Instruction interface of the Hangman game. It provides guidelines on how to play the game. It has a "Back" button that redirects the user back to the Main Menu Interface. This button with a short description of its function is in Table 6.2.

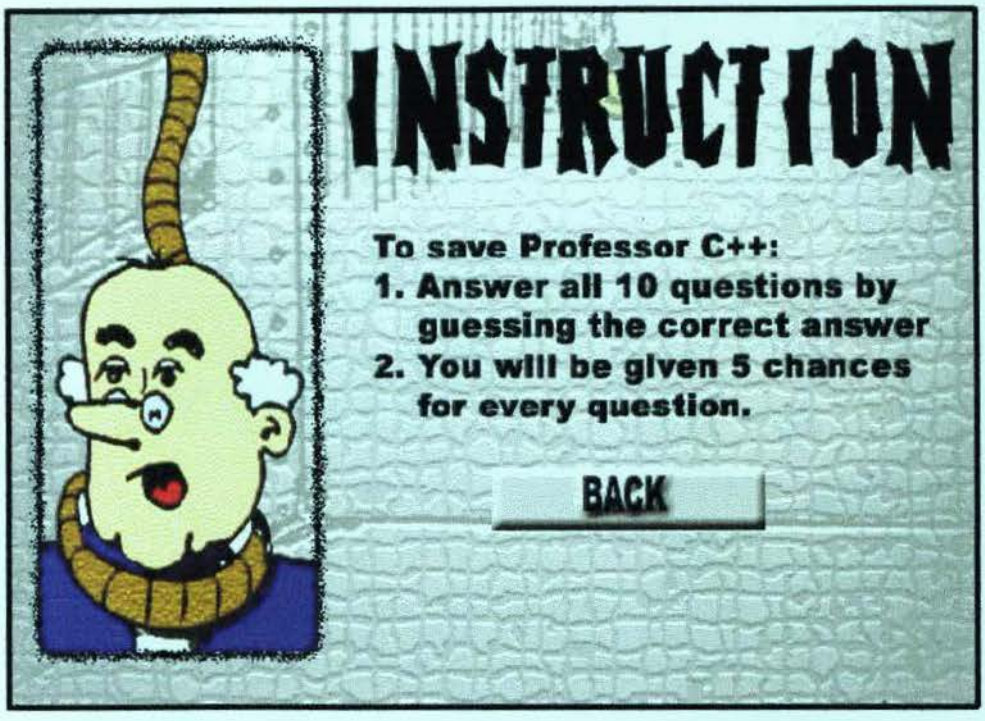


Figure 6.15: Instruction interface of the Hangman game

Table 6.2: Function of Button in the Instruction interface

BACK	When clicked, it will redirect the player back to the Main Menu Interface

- **Game Interface:** This interface shows the rounds of the game. If the player is in Round 1, it is an interface with the word "Round 1", Round 2 with the word "Round 2" and so forth. Figure 6.16 illustrates the Round 1 Game Interface while Figure 6.17 illustrates the Round 2 Game Interface.

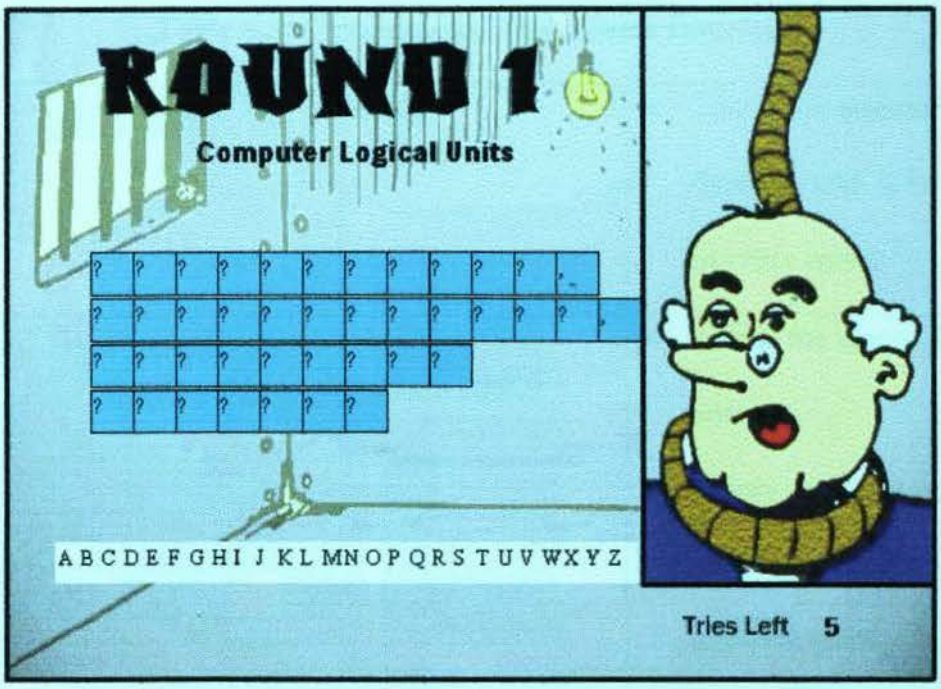


Figure 6.16: Round 1 Game Interface of the Hangman game

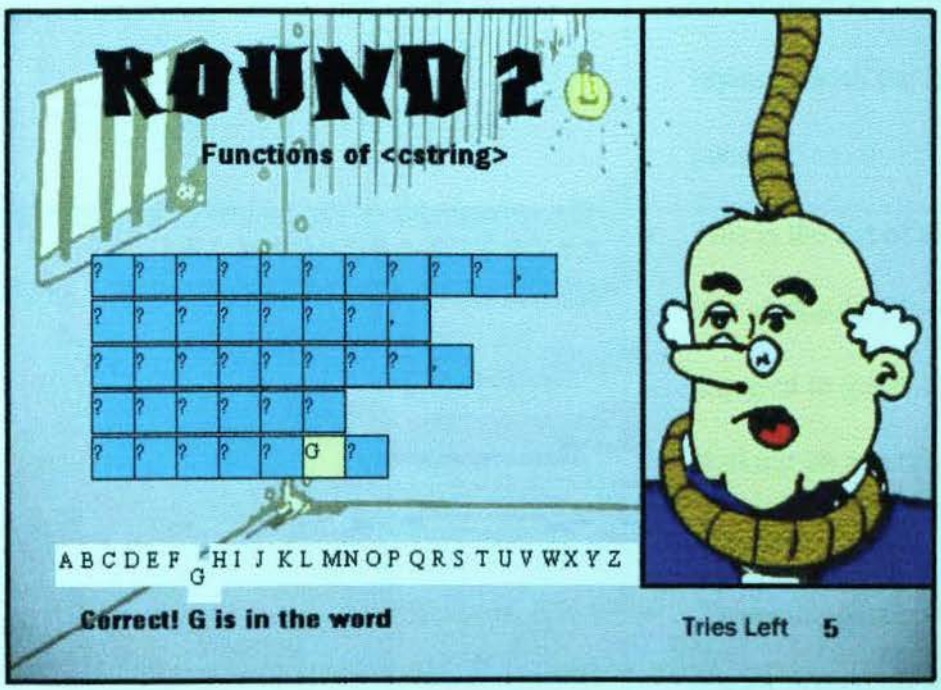
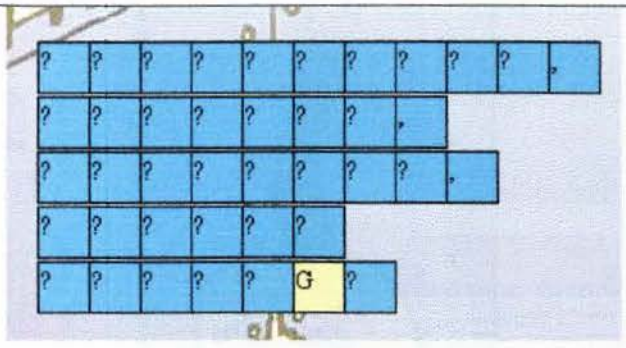


Figure 6.17: Round 2 Game Interface of the Hangman game

The components of the Game Interface are as shown in the Table 6.3.

Table 6.3: Components of the Game Interface

Functions of <cstring>	• This is the question. It is a dynamic text. It is different in every round
	• This is the empty slots for answers. • Empty slots (not guessed) are blue in colour with a question mark. • Correctly guessed slot will be yellow in colour. Question mark will be replaced with the letter guessed.
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	• This is the list of alphabets. • It is clickable. Players are required to guess letters by clicking on a particular alphabet. • Chosen alphabet will be placed in a lower position (eg. look at the letter 'G').
Correct! G is in the word	• This is the feedback field.

	It corresponds with the player's action.
Tries Left 5	• This is the indicator of number of guesses left.
	• This is the movie clip of Professor C++. It is an animation showing the the tightening of the noose on the Professor's neck. Figure 6.18 shows the sequence of the animation • The image of Professor C++ changes according to the number of tries left.

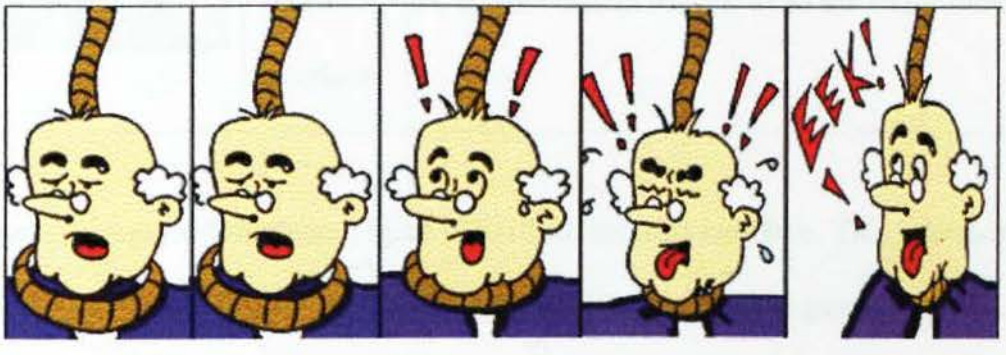


Figure 6.18: Animation of the Professor C++ movie clip

- **Game Over Interface:** Figure 6.19 depicts the Game Over interface. It will be shown when the player has used up all chances in guessing the answer. It has a "Play Again" button. The function and its image will be shown in Table 6.4.

Figure 6.19: Game Over Interface of the Hangman game

Table 6.4: Function of button in Game Over interface

PLAY AGAIN	When clicked, it will redirect the player to the Main Menu Interface

- **Game Complete Interface:** Figure 6.20 illustrates this interface. This interface acts to congratulate the player upon completing all 10 rounds of the game. After 2 seconds, it will automatically direct the user to the Game Over Interface.

Figure 6.20: Game Complete Interface of the Hangman game

6.3.4 Coding of the Hangman Game in Flash

The following subsections display only part of the programming codes of Hangman.

Please refer to Appendix B for the full coding.

6.3.4.1 Preload Interface Coding

```
var lBytes = this.getBytesLoaded();
var tBytes = this.getBytesTotal();

var percentLoaded = Math.floor((lBytes/tBytes) * 100);
loader_mc.bar_mc._xscale = percentLoaded;
loader_mc.percent = percentLoaded + "% of " + Math.floor(tBytes/1024) + "K loaded";

if(lBytes >= tBytes && tBytes>0) {
        if(count>=12) {
                gotoAndStop("main");
        }
        else {
                count++;
```

```
                gotoAndPlay("preload");
        }
} else {
        gotoAndPlay("preload");
}
```

6.3.4.2 Start Game Coding

```
function startGame() {
        newRound();  //reset variables and other assets
        randomNumber = Math.floor(Math.random() * wordSelection.length);
        randomWord = wordSelection[randomNumber];
        wordSelection.splice(randomNumber, 1);
        createWord(randomWord);
        randomQuestion = qSelection[randomNumber];
        qSelection.splice(randomNumber,1);
        question = randomQuestion;
}

function newRound() {
        roundNumber++;
        editedWord = 0;
        readyToPlay = true;
        Rounds.gotoAndStop(roundNumber);
        Face.gotoAndStop(1);

        positionAlphabet();    //create and position the list of alphabets at the bottom
        word_mc._visible = true;
        alphabet_mc._visible = true;
        fieldFeedback_txt.text = "";
        fieldFeedback_txt._visible = true;
        triesLeft_txt._visible = true;
        roundAdv.start();       //play the roundAdv sound clip
}
```

6.3.4.3 Check Game Status Coding

```
function gameStatus() {
        trace("gameStatus() invoked");
        for (var i = 0; i < editedWord; i++) {
                if (this.word_mc["letter_" + i].text == "?") {
                        allMatch = false;
                        break;
                }
                else {
                        allMatch = true;
                }
```

```
    }
    //Winning a round of the game
    if ((allMatch == true) && (Face._currentFrame != 6 )) {
        fieldFeedback_txt = "You won this round!";
        //fieldFeedback_txt.text = "You won this round!";
        reset = true;
    }
    //losing a round in the game
    else if ((allMatch == false) && (Hangman._currentFrame == 6)) {
        fieldFeedback_txt = "You lost. Thanks for playing";
        //fieldFeedback_txt.text = "You lost this round. Thanks for playing";
        reset = true;
        gotoAndStop("lose");
        soundGameOver.start();
    }

    //Winning or completing the game
    if (wordSelection.length == 0 && reset == true) {
        readyToPlay = false;
        fieldFeedback_txt = "Thanks for playing!";
        //fieldFeedback_txt.text = "Thanks for playing!";
        gotoAndStop("Win");
        tada.start();
        word_mc._visible = false;
        alphabet_mc._visible = false;
        fieldFeedback_txt._visible = false;
        triesLeft_txt._visible = false;
        delayAgain = setInterval(createAnotherDelay, 2000);
    }

    //Losing the game
    else if (reset) {
        readyToPlay = false;
        reset = false;
        delayNextRound = setInterval(createDelay, 2000);
        gotoAndStop("game");
        word_mc._visible = true;
        alphabet_mc._visible = true;
        fieldFeedback_txt._visible = true;
        triesLeft_txt._visible = true;
    }
}
```

6.4 COMPUTER SPREE

6.4.1 Introduction

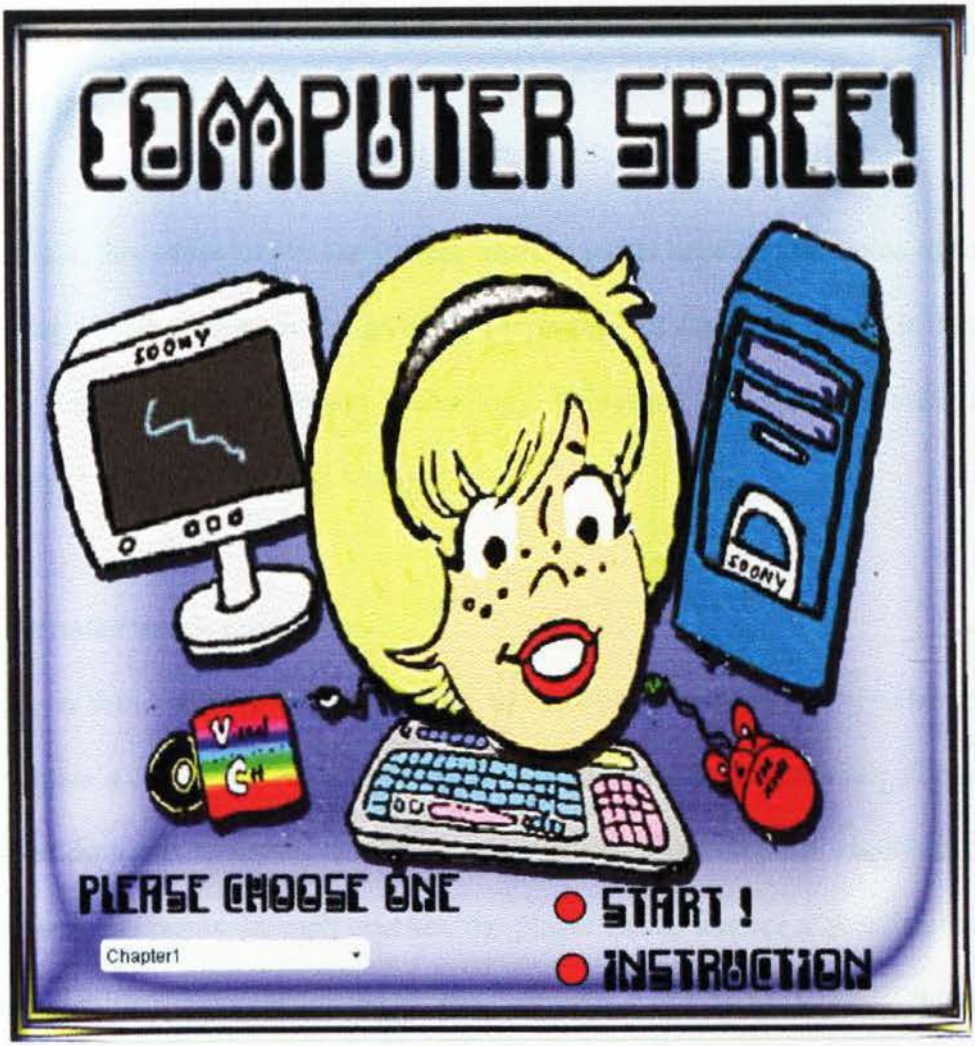


Figure 6.21: Computer Spree Main Interface

Figure 6.21 depicts the main menu of Computer Spree Game.

This game, on the whole, focuses on both theoretical and practical parts of C++ procedural programming. Both parts are portrayed as multiple choices questions.

Players, before starting the game, are required to choose a chapter that they want to focus on. This function enables the user to concentrate on chapters that they are weak in. Thus, achieving one of the objective of this system – enhance understanding of C++ procedural programming.

In actual fact, this game targets the female learners partly because the theme of the game is shopping. Another thing is the growing percentage of female students in Faculty of Computer Science and Information Technology. This game will cater for female learners who found C++ programming is tough.

6.4.2 Computer Spree Overview

This game is about Sabrina who would like to learn programming in C++. However, she does not have a computer. So, she needs help in assembling a computer. The player is required to do some shopping from a computer shop. The computer shop has counters for each item she needs. In each counter, the player has to answer a question to acquire the respective item. Upon acquiring all the computer items Sabrina needs, the player is required to head to the cashier to win the game.

6.4.3 Flow of Computer Spree Game

6.4.3.1 Flowchart of Computer Spree game

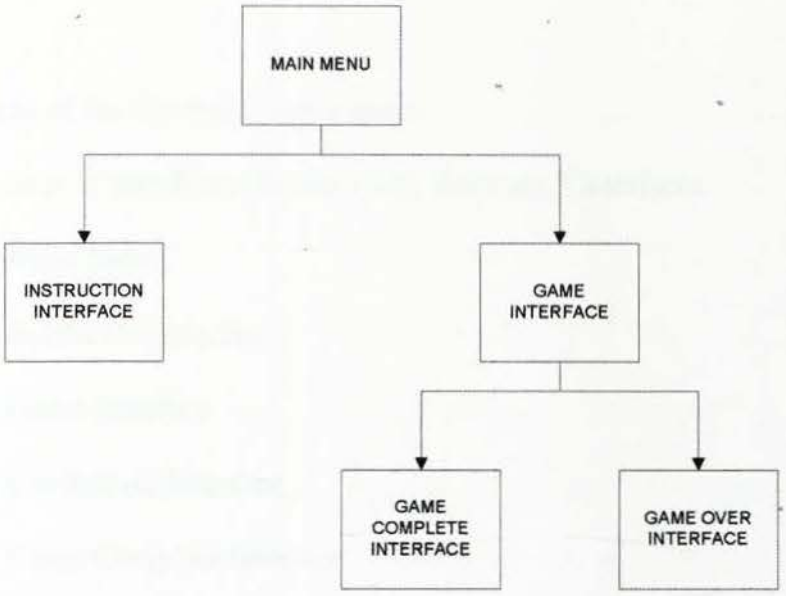


Figure 6.22: Flowchart showing the interfaces in the Computer Spree game

Figure 6.22, as shown above, is a flowchart that shows the flow of interfaces in Computer Spree game. The flow described as follows:

- Main Menu is the first interface as shown in Figure 6.21. This is the introduction page to the game. From this interface, it will either direct the player to the Instruction interface or the Game interface depending on the button clicked.
- Instruction interface will appear if the Instruction button is clicked. From the Instruction interface, it will direct the player back to the Main Menu

- Game interface will appear if the Start button is clicked. From the Game interface, either the Game Over interface or Game Complete interface will become visible depending on the game status.

6.4.3.2 Interfaces of the Computer Spree game

From the flowchart of interfaces (Figure 1.10), there are 5 interfaces:

1. Main Menu
2. Instruction Interface
3. Game Interface
4. Game Over Interface
5. Game Complete Interface.

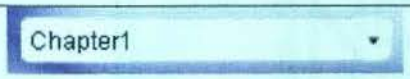
The following subsection will describe each interface in detail.

6.4.3.2.1Main Menu

Main menu, as shown in Figure 6.21, acts as the introductory page to the game. It has few components. Table 6.5 shows the image and description of the components on the Main Menu

Table 6.5: Components in the Main Menu

	• This is the Start button • It will direct the player to the Game interface to start playing the game upon clicked
	• This is the Instruction button • It will direct the player to the Instruction

	Interface upon clicked
	• This is a combo box • It allows the player to choose a chapter of C++ programming. • The questions in the game will reflect the chapter chosen.

6.4.3.2.2 Instruction Interface

Figure 6.23 illustrates the Instruction Interface of the Computer Spree game. Instruction interface contains the general information about the game and guidelines on how to play the game.

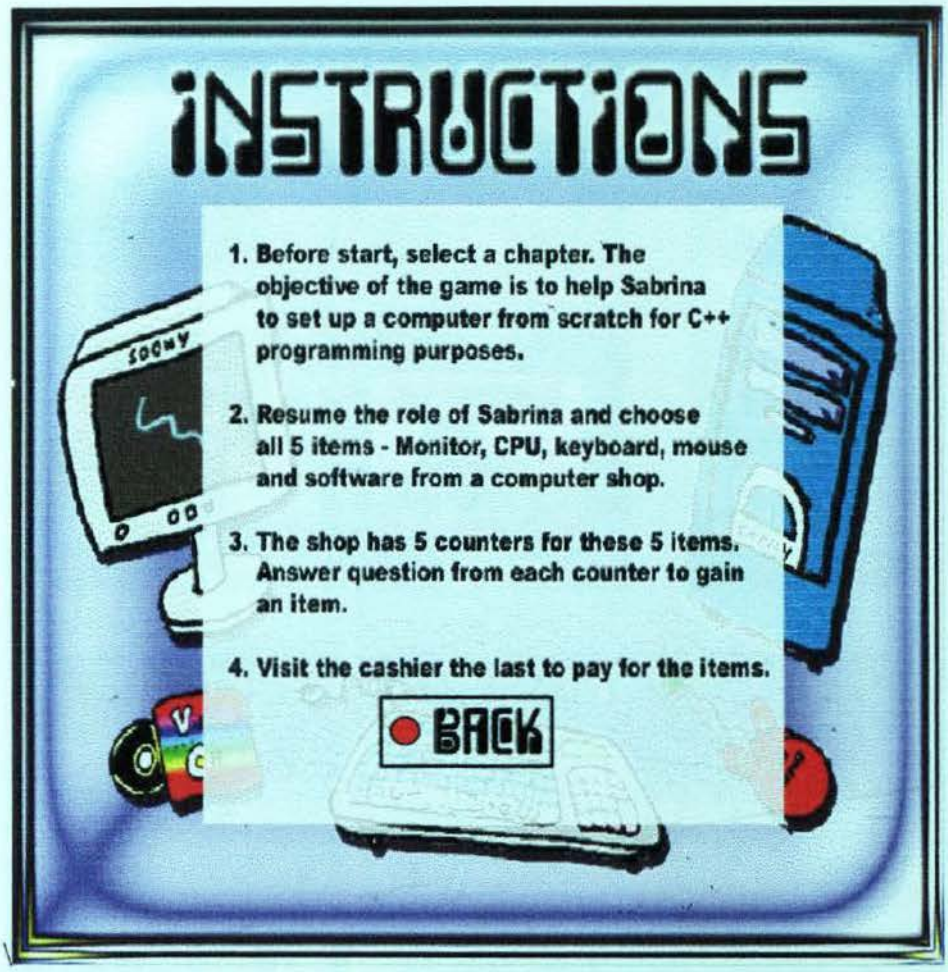


Figure 6.23: Instruction Interface of the Computer Spree game

The interface has a Back button, that upon click, it will redirect the player to the Main Menu.

6.4.3.2.3 Game Interface

Game interface is the main part of the game. Figure 6.24 shows the initial state of the interface. Figure 6.25 shows the state where questions and answer options are popped out.\

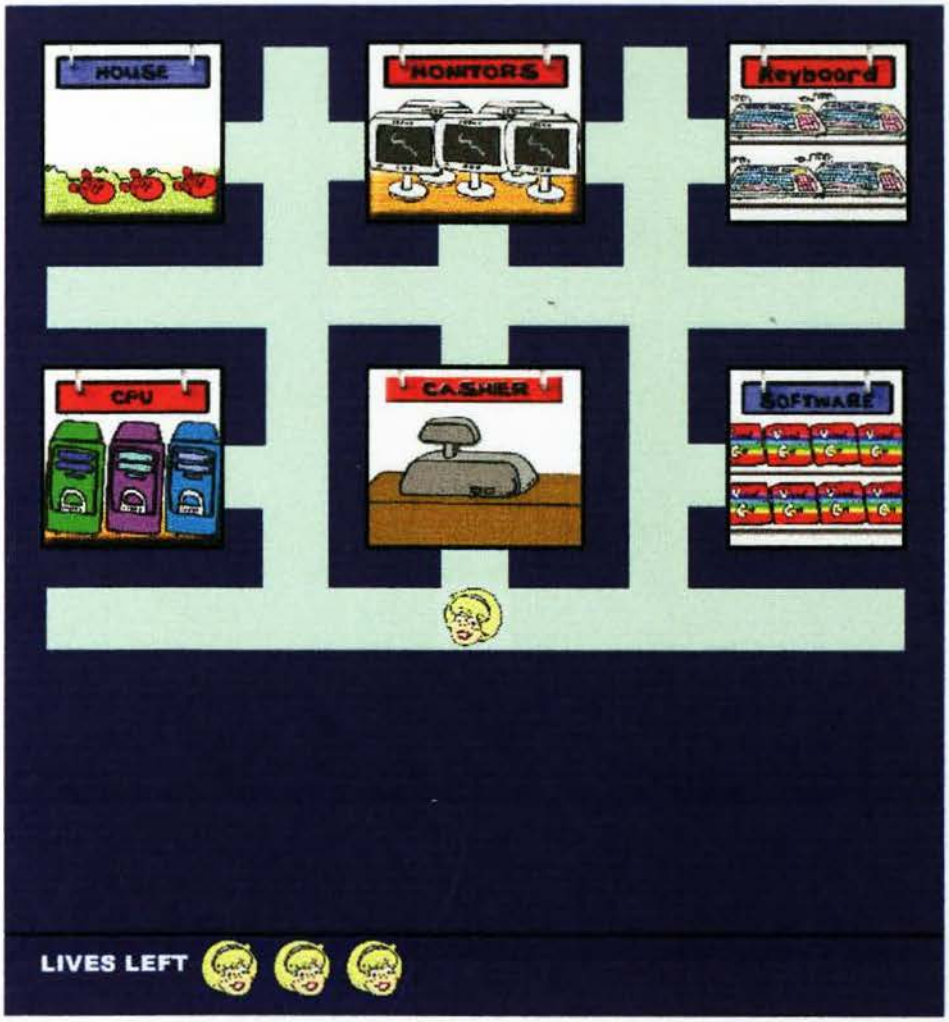


Figure 6.24: Initial Game Interface of the Computer Spree game

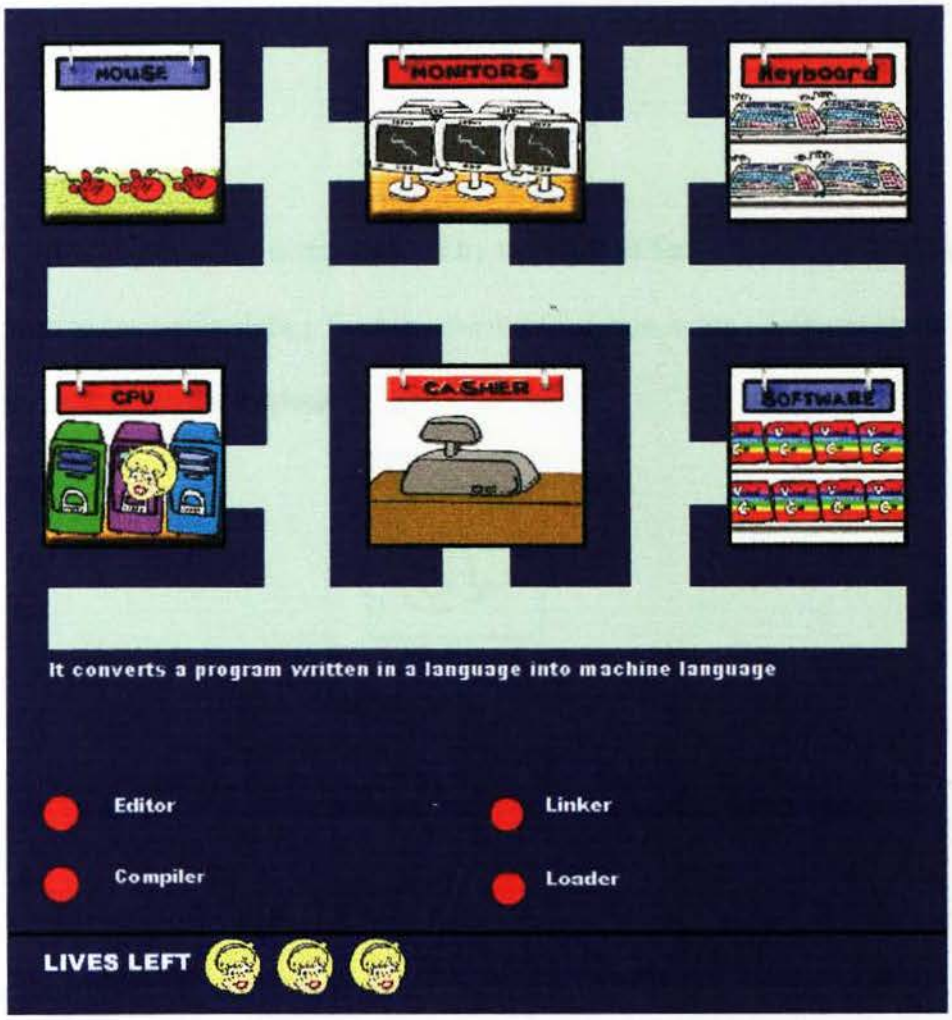


Figure 6.25: Game Interface of the Computer Spree game with Questions and Answer Options

6.4.3.2.3.1 Components in the Game Interface

The components are:

1. Sabrina movie clip (the moving face of Sabrina)
2. Lives Left movie clip
3. Shop counter movie clips (Software Counter, Monitor Counter, CPU counter, Keyboard Counter, Mouse counter)
4. Questions and answer options

5. Feedback Textfield

Sabrina movie clip

This movie clip (Figure 6.26) is represented by the head of Sabrina. The player assumes he/she is Sabrina by manipulating the movement of Sabrina in the computer shop by using the arrow keys in the keyboard.

Figure 6.26 Sabrina

Lives Left movie clip

This movie clip represents the number of chances left for the player during the game. Initially, the player has 3 chances (3 lives), so the Lives Left movie clip will have three Sabrina heads. If the player lost a life, the number of Sabrina heads left will be deducted by one also. This movie clip will continue to until there are no more Sabrina head left. Figure 6.27 shows the sequence of Lives Left movie clip.

Figure 6.27 Lives Left movie clip with description

Shop Counters movie clip

There are 5 shop counters and a cashier counter. Those shop counters are the Software Counter, Monitor Counter, CPU Counter, Keyboard Counter and Mouse Counter. Each of these is a flash movie clip that animates when the player is able to answer the question when Sabrina reaches that counter. Figure 6.28, 6.29, 6.30, 6.31 and 6.32 shows the changes of the graphic when the Software Counter, Monitor Counter, CPU Counter, Keyboard Counter and Mouse Counter animates.

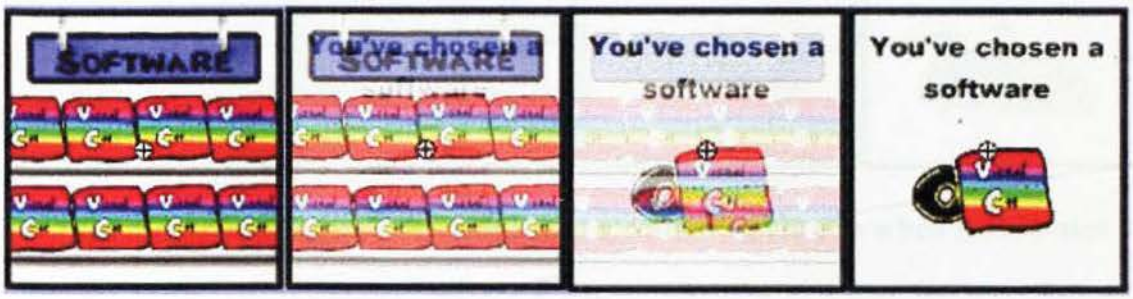


Figure 6.28: Changes to the Software Counter movie clip when it animates

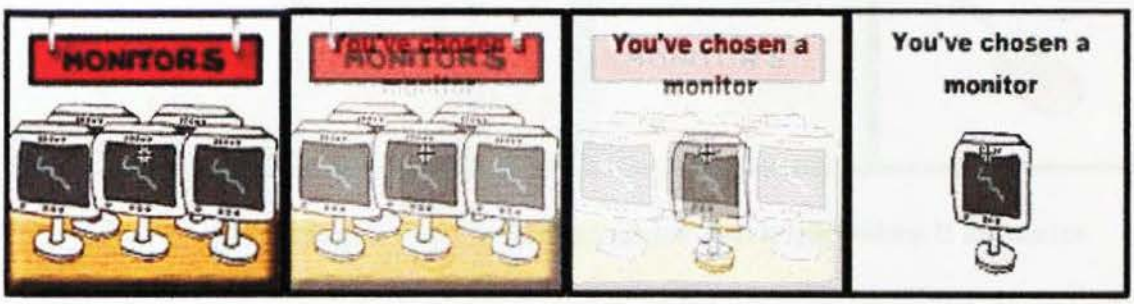


Figure 6.29: Changes to the Monitor Counter movie clip when it animates

Figure 6.30: Changes to the CPU Counter movie clip when it animates

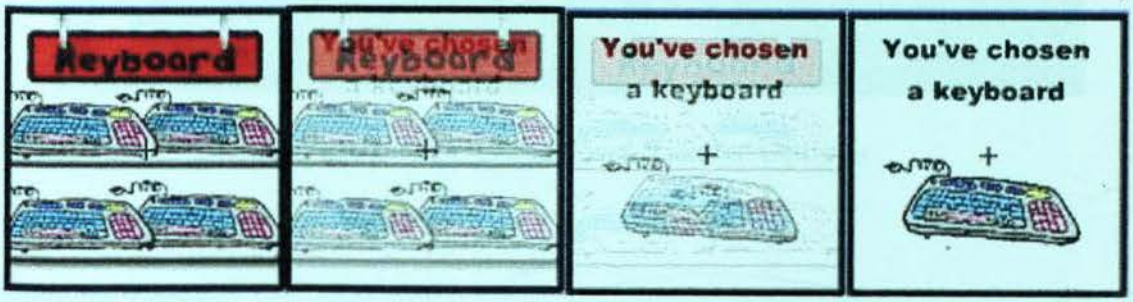


Figure 6.31: Changes to the Keyboard Counter movie clip when it animates

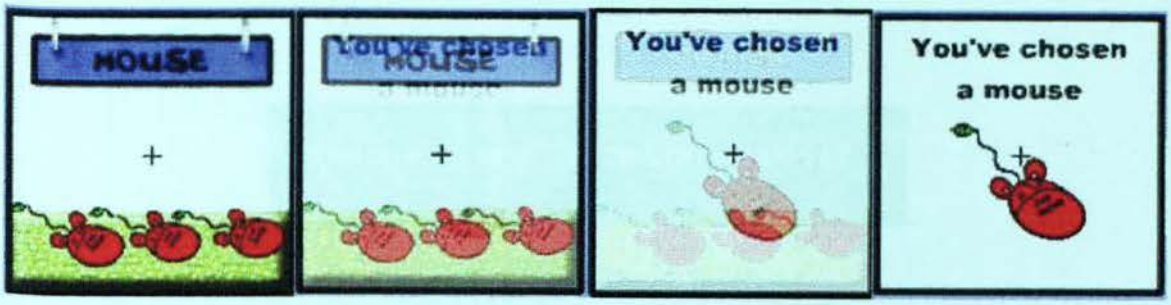


Figure 6.32: Changes to the Mouse Counter movie clip when it animates

Questions and Answer Options

This is the question and answer options as the name indicated. It will only be visible when the Sabrina head moves within the confines of a counter. The player will be required to answer the question by using the mouse to click to select an answer. Figure

6.33 shows the question and answer options component. This content of this component depends on the value of the combo box.

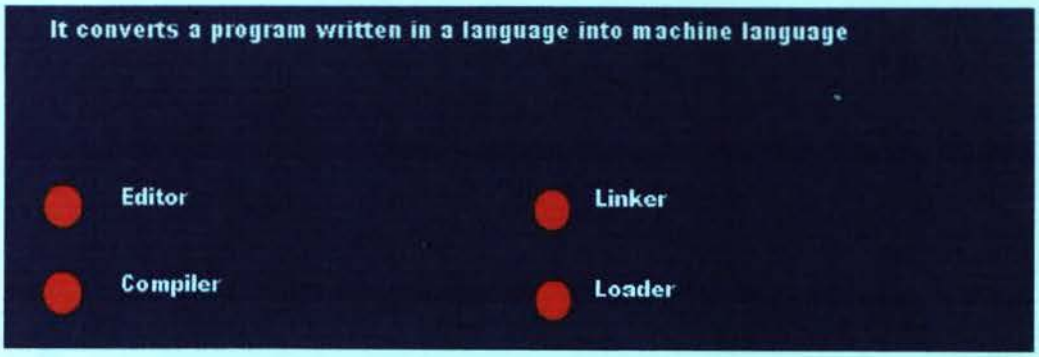


Figure 6.33: Question and Answer Options

Feedback Textfield

Figure 6.34 shows the feedback textfield. It tells the user when if he/she have successfully chosen an item or otherwise.

Figure 6.34: Feedback textfield

6.4.3.2.4 Game Over Interface

This interface will be shown if the player has used up all their chances in the game. It conveys the message that the player has lost and the game is over. On the interface itself, there is a "Play Again" button (as shown in Figure 6.35). When clicked, it will direct the

player back to the Main Menu so that is the player wishes to play again, he/she may be able to do so. Figure 6.36 is the Game Over Interface

Figure 6.35: Play Again Button

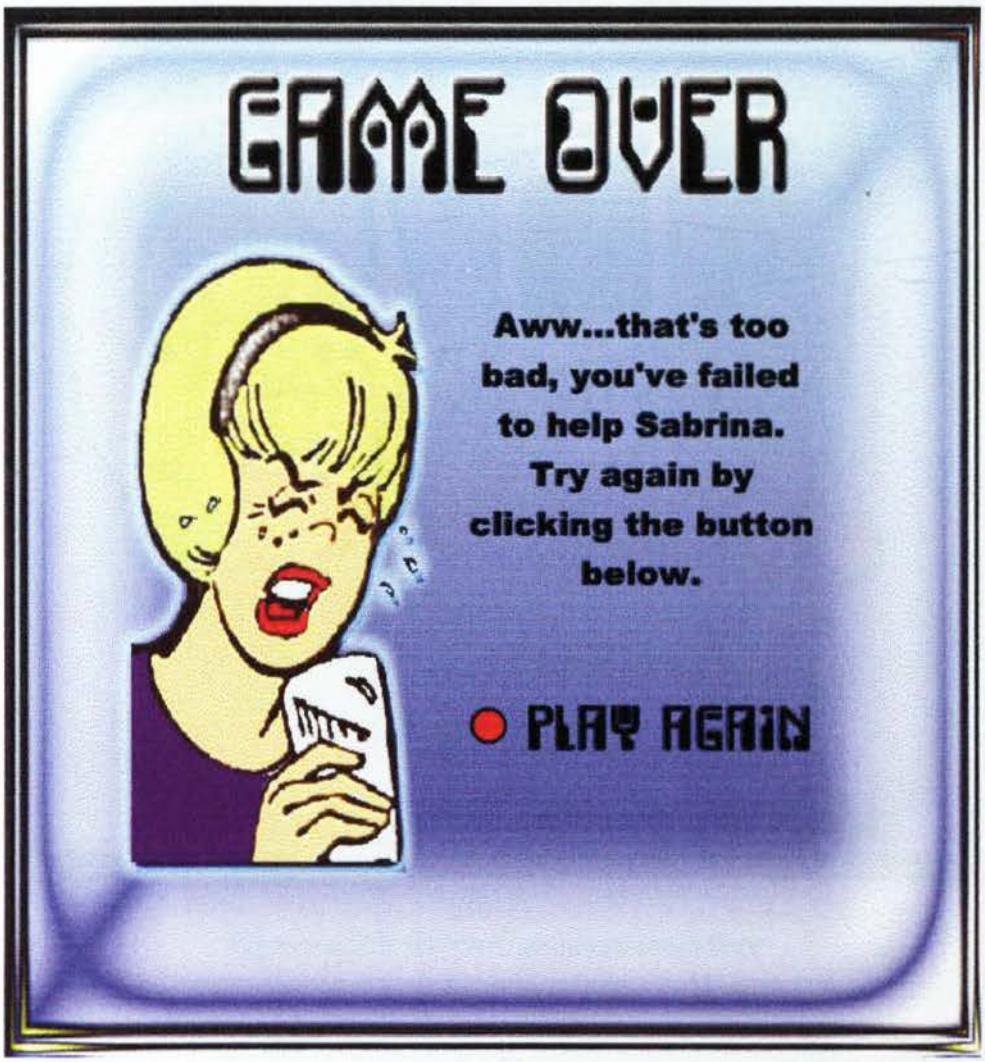


Figure 6.36: Game Over Interface

6.4.3.2.5 Game Complete Interface

This interface is meant to congratulate the player upon answering all the questions in the game correctly. Like the Game Over interface, it has the "Play Again" button as well, doing the same function. Figure 6.37 illustrates the Game Complete interface.

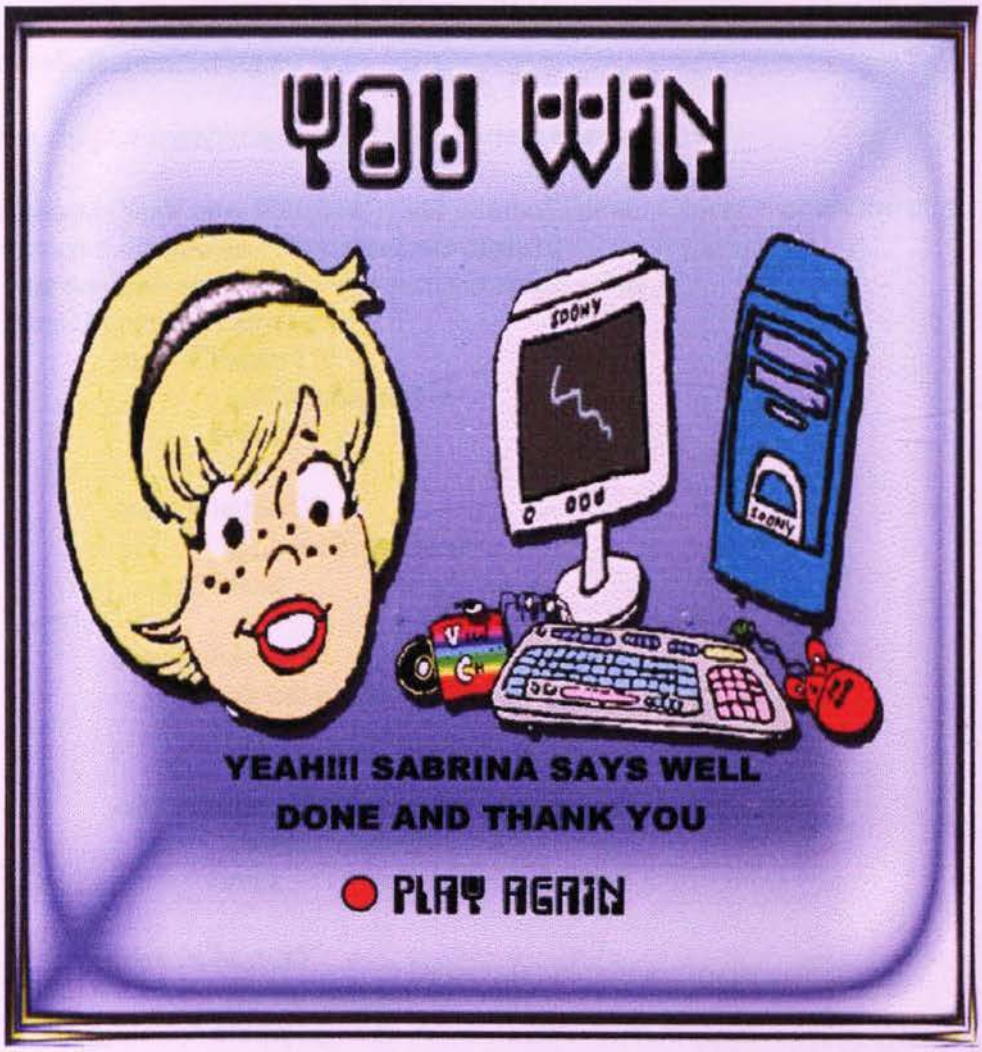


Figure 6.37: Game Complete Interface

6.4.4 Coding in Flash

The following subsections display only part of the programming codes of Hangman.

Please refer to Appendix B for the full coding.

6.4.4.1 Object Handler and Event Listener for the Combo Box (Main Menu)

```
var comboSelection:String = "";

//pass the chapter number selected to the correct chapter array

_global.chapterNo = new Number(); var comboListener:Object = new Object();
comboListener.dropdown = function(eventInfo) {
        var comboValue:String = eventInfo.target.value;
        switch (comboValue) {
                case "Chapter1":
                        _global.chapterNo = 1;
                        break;

                case "Chapter2":
                        _global.chapterNo = 2;
                        break;

                case "Chapter3":
                        _global.chapterNo = 3;
                        break;

                case "Chapter4":
                        _global.chapterNo = 4;
                        break;

                case "Chapter5":
                        _global.chapterNo = 5;
                        break;
        }
};

comboChapter.addEventListnener("close", comboListener.dropdown);

this._lockroot = true;
stop();
```

6.4.4.2 Coding for choosing questions according to chapter chosen

```
//for chapter 1 only if chapter1 is chosen in the combo box
if(_global.chapterNo == 1) {
        for(i=0; i<5; i++) {
                randomQNo = Math.floor(Math.random() * qChap1.length);
                qSelection.push(qChap1[randomQNo]);
                qChap1.splice(randomQNo, 1);
                ans0.push(answer0Chap1[randomQNo]);
                answer0Chap1.splice(randomQNo, 1);
                ans1.push(answer1Chap1[randomQNo]);
                answer1Chap1.splice(randomQNo, 1);
                ans2.push(answer2Chap1[randomQNo]);
                answer2Chap1.splice(randomQNo, 1);
                ans3.push(answer3Chap1[randomQNo]);
                answer3Chap1.splice(randomQNo, 1);
                correctAns.push(correctAnswerChap1[randomQNo]);
                correctAnswerChap1.splice(randomQNo, 1);
        }
}
```

6.4.4.3 Coding to move Sabrina

```
function move() {
        //see whether it is time for another move
        if (getTimer() > sabrina.nextmove) {
                //do not allow another move for 50ms
                sabrina.nextmove = getTimer() + 50;

                //move sabrina
                sabrina._x += sabrina.move.x*5;
                sabrina._y += sabrina.move.y*5;

                //see whether sabrina has reached her destination
                if((sabrina._x == sabrina.dest.x) and (sabrina._y == sabrina.dest.y)) {
                        visitCounter();
                        fieldFeedback_txt._visible = false;
                        nextMove();
                }
        }
}
```

6.4.4.4 Coding that pops up question and answer options

```
if(Math.abs(Math.floor(softwareCounter._x - sabrina._x))<=25 and
```

```
                                        Math.abs(Math.floor(softwareCounter._y
- sabrina._y))<=25) {
                //if have not chosen a software
                if(softwareCounter._currentframe == 1) {
                        setQuestion(0);

                }
                //already chosen a software
                else if (softwareCounter._currentframe == 30) {
                        fieldFeedback_txt = "You've chosen a Visual C++ software!";
                }
        }
```

6.5 WEB DEVELOPMENT

6.5.1 Web Development Tools

Basically, the things needed to develop the web site to make this system web enabled are:

(1)	Operating System	:	Windows XP Professional
(2)	Web Server	:	IIS
(3)	Web development tool	:	Macromedia Dreamweaver MX 2004
(4)	Programming language	:	ASP (active server page), HTML
(5)	Graphics editor	:	Adobe Photoshop Elements

6.5.2 Development Process

6.5.2.1 Development of the Web Site

Most of the web pages are HTML pages only the page to verify user and add new user are ASP pages. I am using frames for each of the HTML pages, so I started off with the development of both the top, left and main frames.

6.5.2.1.1 Top Frame

The Top frame contains the banner of the website. It is depicted in Figure 6.38. On the left hand side of the banner, there sites the logo. The text in the middle in the name of the website, entitled "C++ Online – Learning Site".

Figure 6.38: Top frame of web page

6.5.2.2 Left Frame

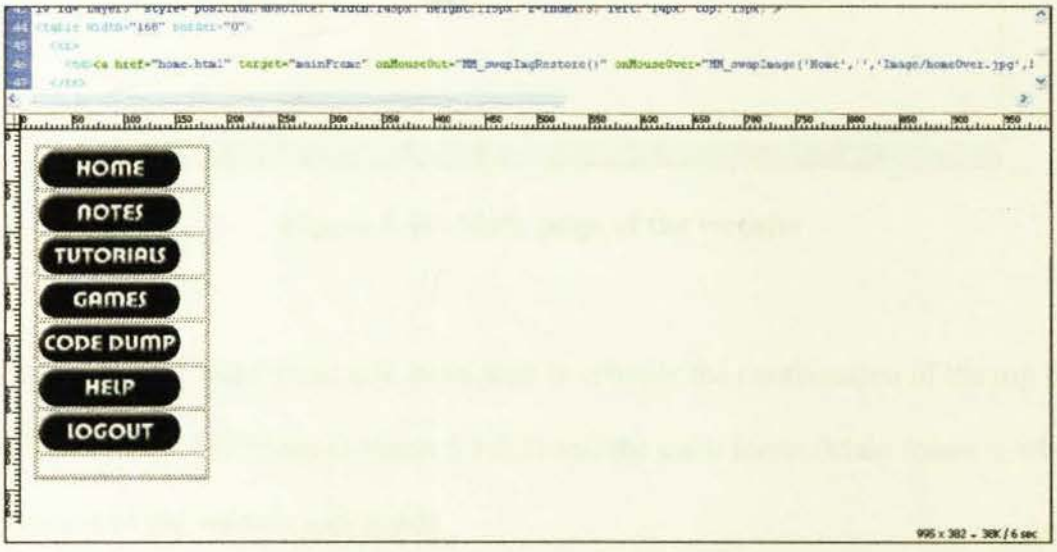


Figure 6.39: Left frame of web page

The Left frame contains the navigational buttons that navigates the user from a page to another. Left Frame is shown in Figure 6.39.

6.5.2.3 Interfaces of the Website

6.5.2.3.1 Main Interface

Main Page will be the default page of the website. It introduces and welcomes the user to the site after the user has login. Figure 6.40 depicts the Main Page of the Website.

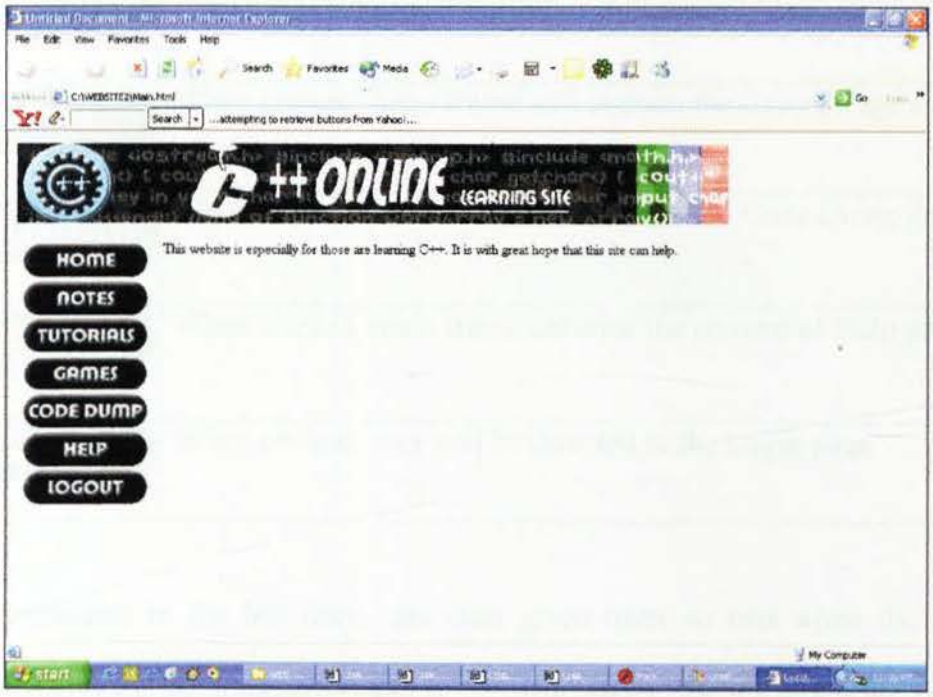


Figure 6.40: Main page of the website

From Figure 6.40, notice that this main page is actually the combination of the top frame (Section 6.5.2.1), left frame (Section 6.5.2.2) and the main frame. Main frame is where the content of the website will reside.

As mentioned in Section 6.5.2.2, the left frame contains the navigational buttons that opens up different pages of the website within the main frame. Table 6.6 shows the function of the each of the button in the left frame.

Table 6.6: Functions of Navigational Buttons in the Left Frame

	When clicked, it will cause the default main page to be output in the main frame (as shown in Figure 6.40)
	When clicked, Notes page will be output in the main frame.
	When clicked, Tutorials page will be output in the main frame.
	When clicked, main frame will contain the Games Page
	When clicked, main frame will contain the Code Dump page
	When clicked, main frame contains the content of Help page
	When clicked, user will be directed to the Login page

Buttons contained in the left frame are then given links so that when the button is clicked it will open the respective page in the main frame. Below is a code sample for one of the buttons.

<img src="Image/home2.jpg" name="Home" width="139" height="39" border="0"></td>

6.5.2.3.2 Notes Page, Tutorials Page, Code Dump Page and Help Page

Figure 6.41 shows the Notes Page of the website. It contains notes that can be viewed within the main frame or downloaded into the computer. Tutorials Page, shown in Figure 6.42 contains tutorial questions and lab exercises that were saved in word format. Code Dump Page, as in Figure 6.43 contains source codes of various topics saved in ZIP format. All these pages has a "Go!" button that automates the downloading process.

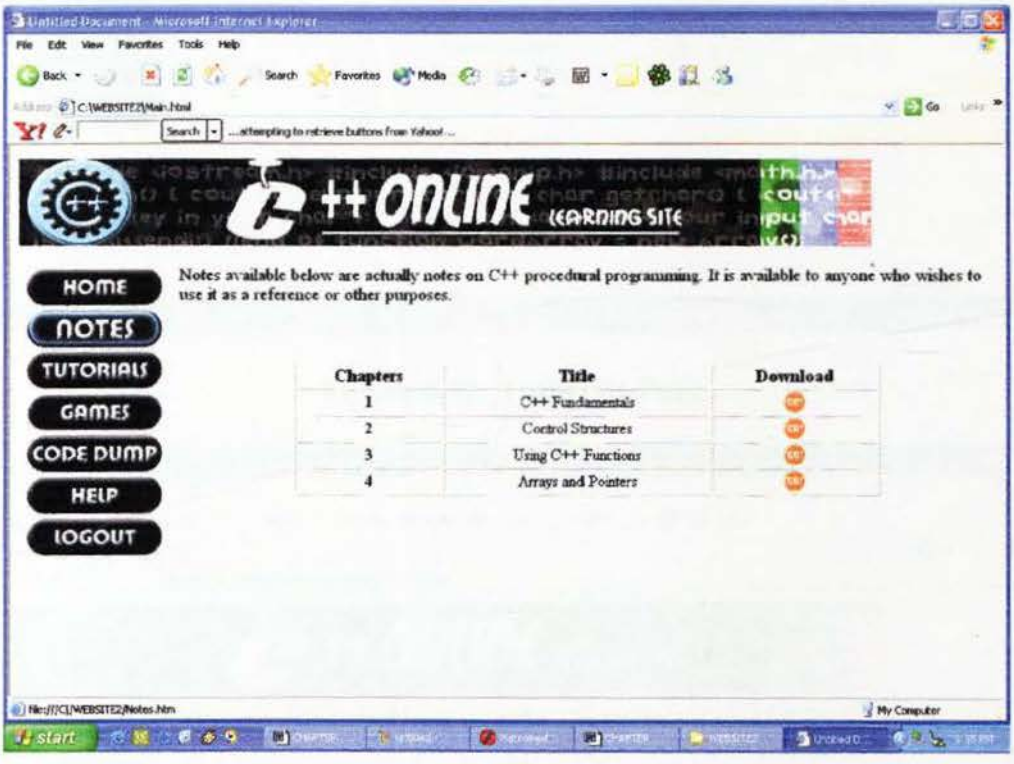


Figure 6.41: Notes Page

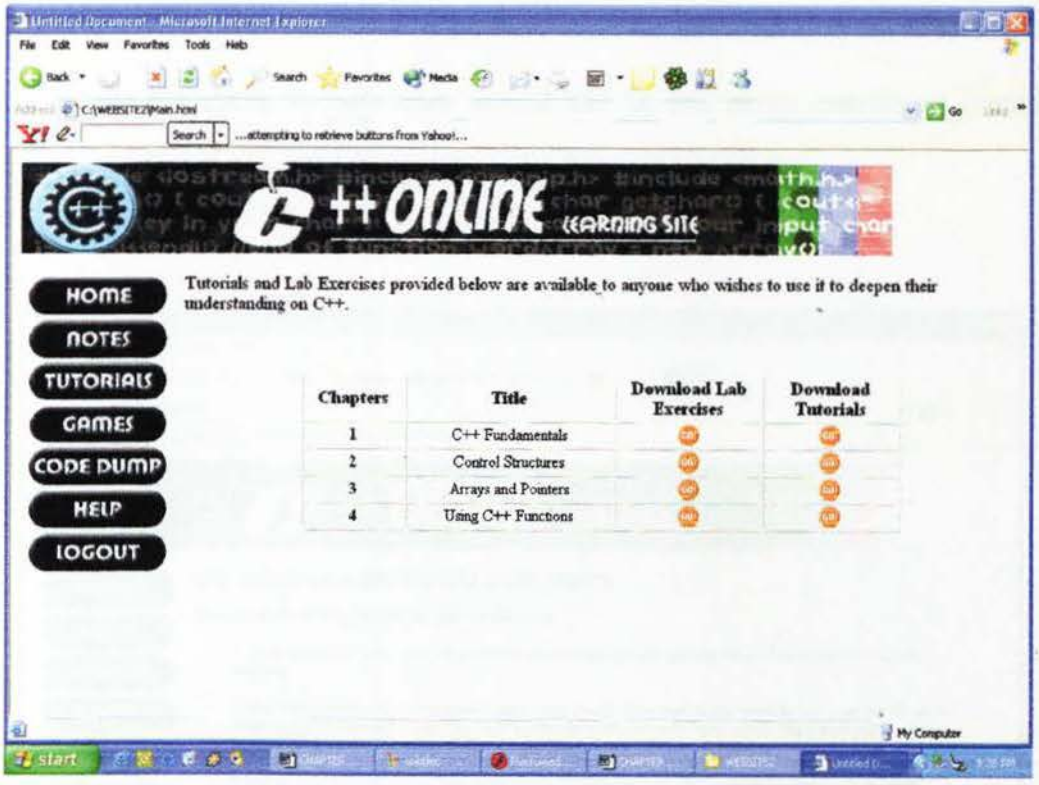


Figure 6.42: Tutorials Page

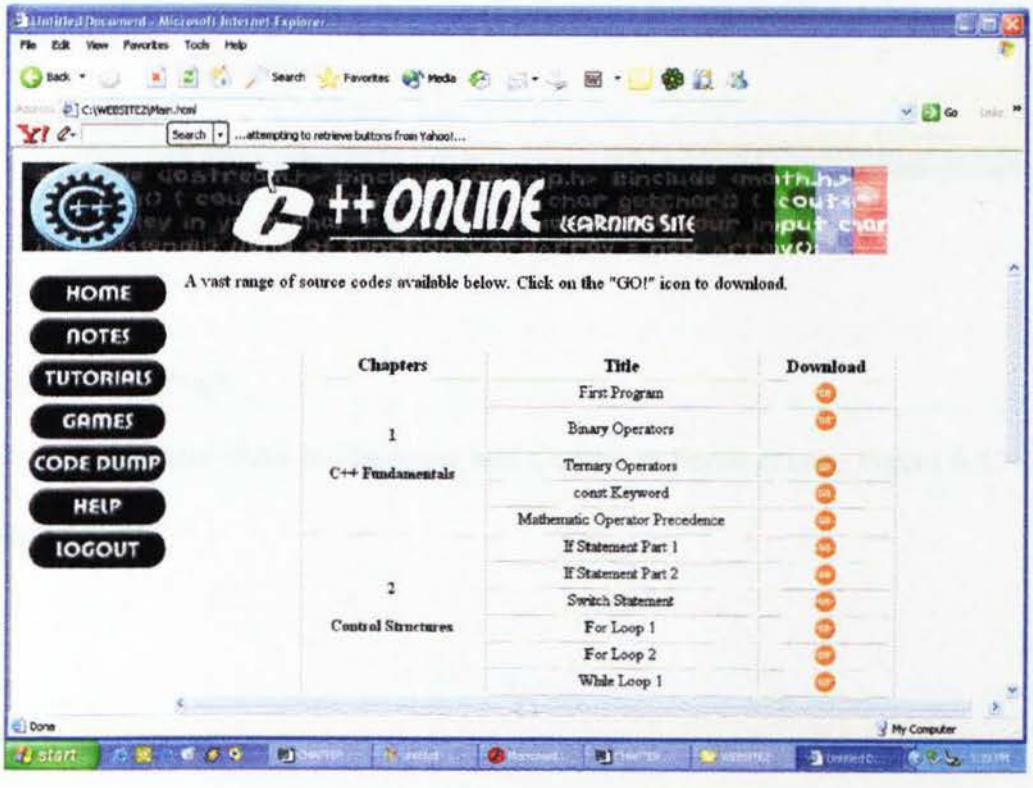


Figure 6.43: Code Dump Page

Help Page contains basic information on how to download and also the administrator's contact details for users, in case user would like to ask make queries or comments. Figure 6.44 shows a screenshot of the Help Page.

Figure 6.44: Help Page

6.5.2.3.3 Games Page

Games Page contains links to Hangman and Computer Spree games. Figure 6.45 shows the Games Page.

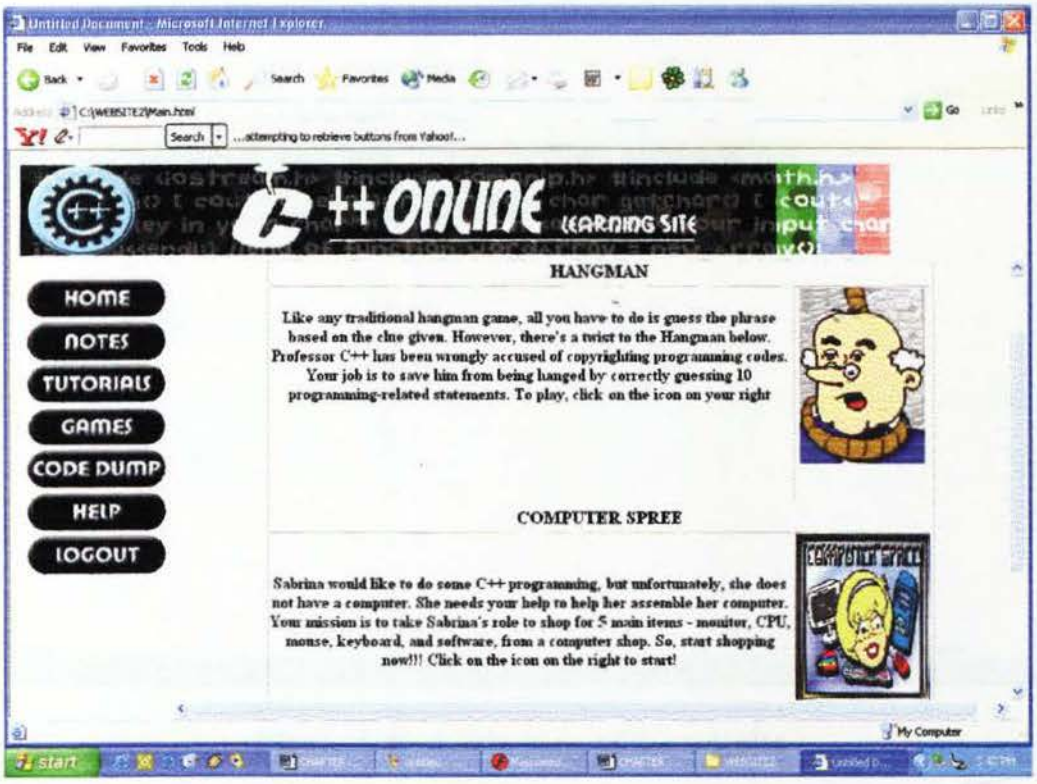


Figure 6.45: Games Page

6.5.2.3.4 Login Page

Figure 6.46 shows the login page of the website. Before the user can actually access the contents of the site, the user has to login and verified by the server. This is done to protect the integrity of the content and privacy of the website as well. To verify and authorize users, ASP codings are used. If the system detected an unauthorized person is trying to access the website, he/she will be redirected to the login page again and again. If he/she wishes to register, a link "New User?" is provided for registration (Figure 6.47).

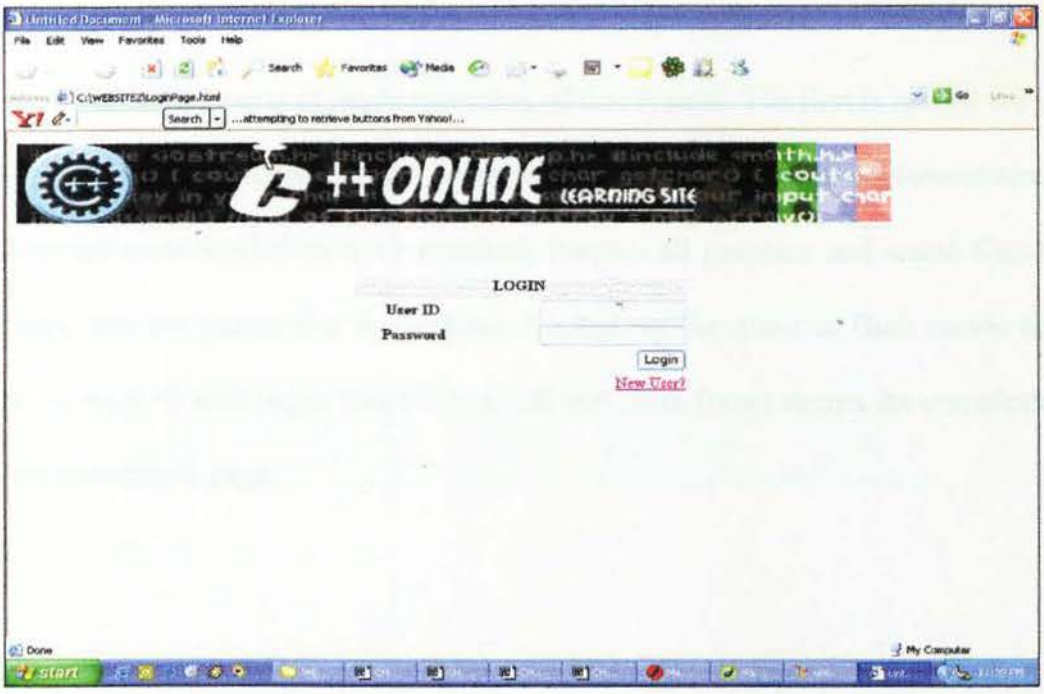


Figure 6.46: Login page of the website

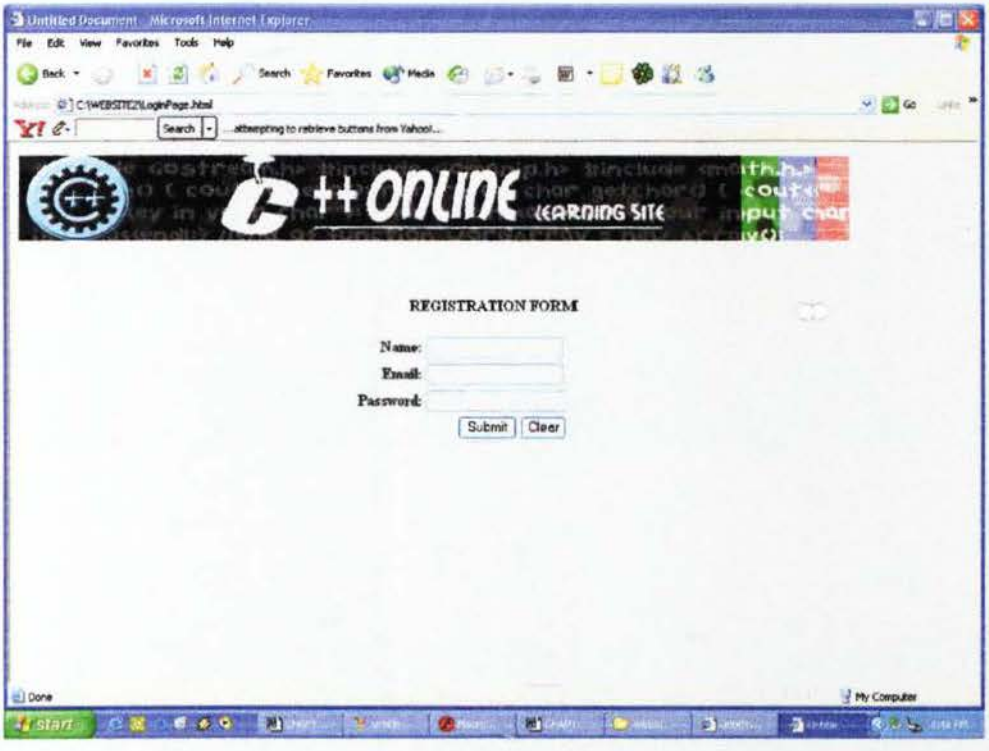


Figure 6.47: Registration page of the website

6.6 SUMMARY

There are basically 2 parts in implementation of this system. The first is related the game development and the second is the website development. Before implementation, all development tools needed must be installed. Prepare all graphics and sound files to be integrated into the games and the website. Publishing the game in flash movie format and integrating all web pages into the top, left and main frame deems the completion of the implementation stage.

CHAPTER 7:

SYSTEM TESTING

AND

EVALUATION

CHAPTER 7: SYSTEM TESTING AND EVALUATION

7.1 INTRODUCTION

In a computer-driven interactive content, mistakes are known more commonly as "bugs" or "glitches". All these bugs or glitches cause problems and errors in the system. That is why it is important to ensure that they are removed. Testing is done on level-by-level basis, beginning with separate units and then finally, the overall combined units. However, testing does not only focus on removing errors, but also at optimization of the system so that it runs in the most efficient way.

The testing method that I employed is black box testing with bottom-up testing strategy. Black box testing focuses on functionality whereby bugs are identified according to its erroneous outputs. If outputs are found to be correct, the internal path of calculations and processing performed are ignored or disregarded. Testing will be done on the games and the websites. All unit testing, integration testing and system testing will be carried out on the games itself and the website.

7.2 TESTING STAGES OVERVIEW

There are 3 stages in testing: (1) Unit testing; (2) Integration testing; and (3) System testing.

Unit testing looks on individual models in the subsystem. Its purpose is to control the number of errors on a small and yet important part of the subsystem.

Integration testing ensures that the all subsystems work together as an integrated unit. Its purpose is to ensure end-to-end flow.

System testing is testing of the application program in isolation after all the subsystems are integrated.

Other than those 3 types of testing, user testing is a must. User testing looks into the effectiveness of the system created in achieving its objectives.

7.3 TESTING OF THE GAME

A game is made of many small entities or objects. All these small entities will combine to form several larger entities. Larger entities when combined will form another larger entity..and so it goes on.

7.3.1 Unit Testing

I started testing by looking at small objects which forms the foundation of both the games I created (unit testing). Small objects include buttons and movie clips.

7.3.1.1 Button

Button is tested by looking at the 4 states of the buttons: Up, Over, Down and Hit. Most of the buttons I use in the game has different image on the 4 frames that depicts the 4 states of the button with respect to the position and action of the mouse.

The steps taken to test a button:

1. Run the test movie command

2. Place the mouse over the button to be tested. Observe what happens
3. Click on the button to be tested. Observe what happens
4. Release the mouse.

If the button acts accordingly to what was planned and designed, then the button is working perfectly. If there are errors, look into the timeline of the button again to find out what is wrong, correct immediately and retest.

7.3.1.2 Movie Clips

Testing of movie clips is only applicable for movie clips that have more than one frame in it. Testing of movie clips is to ensure that the movie clip created animate properly. This can be done by testing the movie clip within the flash environment by pressing the "Enter" key. This will run the movie clip in a simulated environment. I did this mostly to movie clips created for the Computer Spree game. This is because these movie clips have timeline effects on them. Animation can be observed and changes can be made immediately.

Another alternative is to run the game with the test movie command. Wrong and correct inputs were purposely entered to see whether the movie clip will animate in the desired way.

7.3.2 Integration Testing

Integration testing is conducted to see whether these buttons and movie clips, when combined with graphics and sounds will work properly. For example, button actions – it answers the question "When a button is clicked, will the desired output be generated?"

Steps for integration testing (for button):

1. Run the game using the test movie command
2. Click on a button.
3. Observe whether the action will generate the desired output (eg. frame actions along the timeline when the button is clicked)
4. If not, look at the programming code or command of the particular button.

Steps for integration testing (movie clip):

1. Run the game using the test movie command
2. Enter input that will force the movie clip to animate.
3. If the movie clip is animated, then testing is passed
4. If not, review the programming scripts again to detect errors, correct immediately and test again.

Steps for integration testing (frames – ensure that the flow of the game is correct):

1. Press "Enter" to run the movie in the flash environment
2. Observe the flow of the frames from the 1st frame till the last
3. Change immediately if there are any detected incorrect flow.

7.3.3 System testing

After integration testing have been completed, system testing is carried out.

Steps in system testing:

1. Run the game using the test movie command.

2. Start playing the game.
3. Test the game by entering only correct inputs.
4. Observe how the game flows
5. If it flows correct, the movie is then tested with only incorrect inputs. Last of all test with a mixture of correct and incorrect inputs
6. If the movie generates the desired output, the movie can then be published. If not, go back to unit and integration testing to find the errors.

7.4 TESTING OF THE WEBSITE

A website consists of many web pages. Each of these web pages has smaller elements that form the foundation of a particular web page.

7.4.1 Unit Testing

Unit testing is conducted to test the hyperlinks and button links on each individual webpage to ensure that these links will bring the user to the destination page. It is done by clicking on all hyperlinks and buttons and make sure that it will navigate to the correct destination page.

Some buttons however function as a downloading button. Testing is conducted by clicking on the button and ensures that a prompt will be provided for the user to download or open the file within the website. It is important to ensure that the button will download the correct file.

7.4.2 Integration Testing

Integration testing is done after all the individual pages were combined to form the website. The whole website and its files are placed in the IIS folders. Test the whole website in the localhost. It is to ensure consistency among these individual web pages, links to each web page is correctly implemented, and all web pages are linked and work properly.

The most important part is the testing login module of this system. Dummy data are entered into the database. Then, enter the dummy data in the login page to check whether login is permitted.

7.4.3 System testing

The website and its supporting files are then uploaded in the web host. System testing is done by accessing the website through the URL or address of the website. Test the login function and all other functions implemented by the website. Ensure the download function of files (especially games) are functioning properly.

7.5 OTHER TESTS

7.5.1 Testing downloading performance of Flash movie

A report can be generated by Flash to find frames that slows the playback of the movie. It also shows what the Flash has taken to optimize of the movie. This can be done using "Select Generate Report" option during the movie publishing process in Publish Settings dialog box.

7.5.2 Performance testing

This is another test for the created games. The flash movies are published as stand alone projector file or flash movie. Test these published games in actual working environment (in Stroustrup Lab) to ensure it works well.

7.5.3 User Testing

Some errors go undetected when the system is being tested by the developer themselves. So, end users are selected to test the system to detect those errors.

7.6 SYSTEM EVALUATION

System evaluation is conducted to get feedback for eventual improvement and advancement of the system implemented. Three end users were selected to test the system and interviewed to get their opinions and comments. Interview questions will be available in Appendix C.

7.6.1 Result of end user evaluation

The evaluation is strongly focused on the games rather than the website itself. Below is the result of the evaluation.

Q.1 What do you think of the scope of the games? Is it sufficient to learn C++?

Interviewees said that the scope is significant in Shopping Spree game but rather loosely defined in the Hangman game. However, each fulfils its own purposes. Hangman is more towards recapturing what was learned throughout procedural programming while Computer Spree focuses on chapter-by-chapter learning.

Q.2 Is the design attractive? Does it make learners more interested in playing the game and learning at the same time? Is it suitable for learners at university level?

In general, interviewees said that the games are interesting especially the graphics but colours are quite dull. On the suitability issue, they said most probably because most people who are learning the programming language can be considered as young adults. Young adults prefer games to books.

Q.3 What do you think of the content of the games? Are the questions in the games appropriate?

Interviewees commented that the questions in the Hangman game are theoretical. It does not actually help in learning how to write programs in C++ but serve as a general knowledge to users. This they say can be a good thing because these are important to build a good foundation to learn other programming languages.

Questions in Computer Spree motivate the user to think rather than just guessing words. Some questions require the user to search for the answer.

On the appropriateness of questions, most questions are appropriate.

Q.4 Are the games effective in learning C++?

On the whole, the games are effective in learning the theoretical part of C++. One of the interviewee said that though it does not help in writing out programs but it does help in testing our understanding in programming. This is more important as without a clear idea of the theory of programming, one simply cannot produce a good program.

Q.5 What do you think of the website? Is the design attractive? How about the colour combination?

All interviewees said that the website is simple and attractive. Interviewees commented on the buttons, saying that the rollover effects in the buttons are excellent navigation indicators to users of website.

Q.6 Is the website easy to navigate and use?

All interviewees said yes. Buttons for navigating between web pages and downloading the content of the web pages are provided abundantly.

Q.7 Please give comments and/or suggestions on the system

Interviewees commented that the words in the Computer Spree are small. They would like to see a larger font size to increase readability. Other than that, in the Hangman game, they would also like to guess the letters by pressing keys on the keyboard, rather than just clicking with the mouse.

CHAPTER 8:

CONCLUSION

CHAPTER 8: CONCLUSION

8.1 INTRODUCTION

On the whole, this system achieves the objectives and expected outcome set for this project, although it came with a few constraints. Nevertheless, this system is a reflection of my efforts I poured on this project. I gained lots of knowledge which will be valuable in my working life.

8.2 PROBLEMS FACED DURING DEVELOPMENT AND SOLUTIONS TO IT

In the first part of the thesis, most problems were faced during analysis stage. This is where I have find to the most suitable platforms and software to be used for implementation, appropriate development life cycle and methodology to be used.

In the implementation and testing stage, these are the problems I faced:

- Insufficient knowledge of Macromedia Flash.
    - This is the first time I dabbled with Flash. I have vague idea on how to use Flash and got very confused with the layer and timeline concept
    - This, however, is partially solved by finding tutorials online and buying a reference book on Flash.
- Choosing appropriate images for both the games and website
    - To solve this, I produce my own image and scanned it into the computer to be further processed and manipulated.
- Developing suitable and attractive layout and storyline of the game

- The game should be not too childish but suitable for everyone who is willing to learn C++
- Decided on Hangman, a game played by people of all ages and a game that has adventure sense in it.

- Testing the whole system and integrating the flash in the website

8.3 PROJECT STRENGTH AND LIMITATIONS

8.3 Project strength

The project strengths are:

- Interactivity and user friendly
  - It is found that human learn better when they are captivated and motivated. Learning C++ with games involves direct interaction between the learner and the system.
  - Thus, achieves the objective of motivating the learner to be more keen in learning C++
- Ease of use
  - The system provides sufficient and unambiguous buttons and links that enables the learner to easily navigate through the website
  - With the help instruction in the website, they gain information on how to download the contents of the website
  - With the instruction in the game itself, it helps the learner to quickly learn about what the game is all about.
- Attractive features in the games

- Incorporates sound and animation that captures the attention of the learner
- Has rollover buttons, to let the user knows where to click

- Games have questions that enables C++ learning although more on the theoretical part of C++ programming
  - Computer spree game enables the user to select which chapter they would like the game to focus on. So this achieves the expected outcome that the game will be able to help in chapters of programming that they are weak in. Therefore, it will increase their performance and get a better idea of C++ programming.
  - It also achieves the objective of providing aid to learn concepts of programming in C++.
- The website is user friendly and attractive that gives the learner the freedom to either download notes, tutorials, source codes for future reference.

8.3.2 Project Limitations

The project limitations are:

- No score keeping in any of the games
- No search function in the website
- Limited number of questions in the games
- Games does not really help in writing out programs, instead it focuses on learning the theoretical part of C++ programming
- Only two games are provided in learning procedural C++

8.4 FUTURE ENHANCEMENT

The system could be further enhanced by:

- Implementing a scoring system for the game so that the progress of the learner can be tracked
- Add search function within the website to make searches of items related to a particular C++ keyword easier
- Add more games which are oriented toward writing out programs.

REFERENCES

(Ahlers & Garris, *unpublished*)
Ahlers, R. and Garris, R. (unpublished). Internal presentation: results to be published. Cited in Prensky, M. 2001. *Digital game-based learning.* New York: McGraw-Hill.

(Bass, *unknown year*)
Bass, R. (*unknown year). A Brief Guide to Interactive Multimedia and the Study of the United States.* www.georgetown.edu/faculty/bassr/multimedia.html.

(Boehm, 1988)
Boehm. W. B. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer 21,* May 1988, pp. 61-72

(Brownfield & Vik, 1983)
Brownfield, S. and Vik, G. (1983). Teaching basic skills with computer games. *Training and Development Journal.* **37(12)**.53-56.

(Bruce, 1990)
Bruce, C. S. (1990). Information skills coursework for postgraduate students: investigation and response at the Queensland University of Technology. *Australian Academic & Research Libraries.* **21(4)**, 224-232

(Centurion System Corporated, 1997)
Centurion Systems Incorporated. (1997). I*What is Interactive Multimedia CBT? http://www.centurionsys.com* Last updated: August 07, 1997. Centurion Systems Incorporated.

(Dillon & Leonard, 1998)
Dillon, P. M. and Leonard, D. C. (1998). *Multimedia & the Web from A to Z.* 2nd Edition. Arizona, Oryx Press.

(Feldman, 1994)
Feldman, T. (1994). *Multimedia.* London, Chapman & Hall.

(Hillman, 1998)
Hillman, D. (1998). *Multimedia Technology & Application.* London, England: Delmar Publishers.

(Hodges & Sasnett, 1993)
Hodges, M. E. and Sasnett, R. M. (1993). *Multimedia Computing: Case Studies from MIT Project Athena.* New York, Addison-Wesley.

(Kennedy *et al,* 1982)

Kennedy, R. S., Bittner, A. C., Harbeson, M. M. and Jones, M. B. (1982). Television computer games: A "new look" in performance testing. *Aviation, Space and Environmental Medicine*. **53(1)**.

(Kotonya & Sommerville, 2003)
Kotonya, G. and Sommerville, I. (2003) *Requirements Engineering: Processes and Techniques*. England, John Wiley & Sons, Inc.

(Kramer, 2000)
Kramer, W. (2000). What is a Game? *The Games Journal*

(Locatis, *unknown year*)
Locatis, C. (*unknown year)*. An Interactive Technology Primer. *http://tlc.nlm.nih.gov/*

(Macauley, 2004)
Macauley, P. (2004). *The Literature Review.* *http://www.deakin.edu.au/library/findout/research/litrev.html*. Last updated: May 20, 2004. Deakin University Library.

(McKillup *et al*, 2000)
McKillup, S., Webster, B., and Ward, T. (2000). *http://www.library.cqu.edu.au/litreviewpages/what.htm*. Central Queensland University Library

(Menn, 1993)
(Monkey Wrench Conspiracy, 1999)
Monkey Wrench Conspiracy. (1999). *The Monkey Wrench Conspiracy (how to get engineers to learn and to like it)*.

(Prensky, 2001)
Prensky, M. (2001). *Digital Game-based Learning*. New York: McGraw-Hill.

(Ravenscroft, 2002)
Ravenscroft, A. (2002). Developing and evaluative dialogue games for collaborative e-learning. *Journal of Computer-Assisted Learning*. **18(1)**. 93-101

(Royce, 1989)
Royce, W. W. (1989). Managing the Development of Large Software Systems: Concepts and Techniques, in, *1970 WESCON Technical Papers, Western Electronic Show and Convention, Los Angeles, August 1970,* pp. A/1-1-A/1-9; reprinted in: *Proceedings of the 11th International Conference on Software Engineering* (Pittsburgh, May 1989), IEEE Computer Society Press, Los Alamitos, CA, pp. 328-38.

(Schach, 2005)
Schach, S. R. (2005). *Object-Oriented & Classical Software Engineering*. 6th Edition. New York, McGraw-Hill.

(Schuman, 1998)
Shuman, J. E. (1998). *Multimedia in Action.* Belmont CA, Wadsworth Publishing Company

(Scown *et al*, 2000)
Scown, B., Ward, T., Reaburn, P. and McKillup, S. (2000). *http://www.library.cqu.edu.au/litreviewpages/why.htm.* Central Queensland University Library.

(Sinan Si Alhir, 2002)
Sinan Si Alhir. (2002). *Understanding the Unified Process.* *http://home.earthlink.net/~salhir*

(Stair & Reynolds, 2001)
Stair, R. M. and Reynolds, G. W. (2001). *Principles of Information Systems: A Managerial Approach.* 5th Edition. London, Course Technology.

(Starbuck & Webster, 1991)
Starbuck, W. H. and Webster, J. (1991). When is play productive? *Accounting.* **1(1).**

(Taylor, 2004)
Taylor, D and Procter, M. (2004). *The Literature Review: A Few Tips On Conducting It.* *http://www.utoronto.ca/writing/index.html.* Last Updated: May 11, 2004. University of Toronto.

(Wood & Steward, 1987)
Wood, L. E. and Stewart, P. W. (1987). Improvement of practical reasoning skills with a computer game. *Journal of Computer-Based Instruction.* **14(2).** 49-53.

(UniGame, 2003)
UniGame. (2003). UniGame: Game-based Learning in Universities and Lifelong Learning. *Project Workplan,* "UniGame: Social Skills and Knowledge Training", Minerva project. Retrieved 18.08.2003, from *http://www.unigame.net*

BIBLIOGRAPHY

Books:

Chapman N. and Chapmen J. (2000). *Digital Multimedia*. England: John Wiley & Sons, Inc.

Connolly, T. M. and Begg, C. E. (2002). *Database System: A practical approach to Design, Implementation, and Management*. 3rd Edition. England: Pearson Education Limited.

Olsen, G. (1997). *Getting Started in Multimedia Design: A designer's guide to nutes and bolts of developing multimedia concepts.* Cincinnati, Ohio; North Light Books.

Sebesta, R. W. (2002). *Concepts of Programming Language*. 5th Edition. Boston: Addison-Wesley.

Whitten, J. L., Bentley, L. D. and Dittman, K. C. (2004) *Systems Analysis and Design Methods*. 6th Edition. New York, McGraw-Hill.

Reindhardt, R. and Dowd, S.(2004). *Macromedia Flash MX 2004 Bible.* Indianapolis, Indiana; Wiley Publishing Inc.

Rosenzweig, G. (2003). *Macromedia Flash MX ActionScript for Fun & Games.* Que Publishing.

Bell, D. (1997). *The Essence of Programming Using C++*. Hertfordshire; Prentice Hall Europe.

Dale, N. (2000). *A Laboratory Course in C++*. 2nd Edition. Toronto, Canada; Jones and Bartlett Publishers.

Conference Proceedings, Journals and Magazines:

Bramucci, R. S. (2002). Digital Game-based Learning In Webct, in *WebCT 2002 – 4th Annual User Conference,* Boston, MA.

Gee, J. P. (2004). Learning by Design: Games as Learning Machines. *Interactive Educational Multimedia.* **8.** 15-23

Websites:

http://www.gurukulonline.com/eLearning_whatis.asp

http://www.learningshop.co.uk/introduction.asp#anchor104391

http://www.quasar.ualberta.ca/edpy485/mmedia/educ.htm#intro

http://www.usu.edu/sanderso/multinet/index.html

APPENDIX A

UNIVERSITY OF MALAYA
FACULTY OF COMPUTER SCIENCE
AND
INFORMATION TECHNOLOGY

I am currently developing web-based software to learn C++ programming language through gaming. This questionnaire is about learning C++ programming language through multimedia educational games. It is to help to deepen my understanding on introducing game-based learning of C++ programming language. Kindly answer all the questions appropriately and as honest as possible. Thank you for your cooperation. Your help is deeply appreciated.

(A) Demography

1. Faculty: _____ (eg. FSKTM)

2. Course: _____ (eg. Software Engineering)

3. Year of intake: _____ (eg. 2002/03)

(B) Game

The following questions require you to answer using the scale given below:

Scale	1	2	3	4	5
	Very Easy	Easy	Moderate	Hard	Very Hard

4. Rate the toughness for every sub-topic of C++					
a) Basics of programming	1	2	3	4	5
b) Control Statements	1	2	3	4	5
c) Functions	1	2	3	4	5
d) Arrays	1	2	3	4	5
e) Pointers	1	2	3	4	5

5. Which category of multimedia games is relevant to learn C++?
   - ☐ Puzzle Games
   - ☐ Word Games
   - ☐ Answer-and-Questions (Quiz-like games)
   - ☐ Adventure Games
   - ☐ Others (Please specify: _____)

6. For every type(s) of multimedia game(s), please give example(s).

_____

7. How would you like a game to help you?
   - ☐ Understand definitions
   - ☐ Understand concepts
   - ☐ Write short codes or program
   - ☐ Others (Please specify: _____)

8. What kind of approach should be adopted by the game?
   - ☐ Notes approach (games will be like a slide show)
   - ☐ Tutorials approach (games will let user answer questions)

9. What type of game would you expect to see?
   - ☐ Chapter specific game
   - ☐ Combination of chapters game

10. If it is a combination of chapters, how you would like the content of each chapter to be arranged in the game?
    - ☐ Follow chapters in ascending order
    - ☐ Sort chapters according to level of toughness
    - ☐ Random

11. How would you like to navigate through the game?
    - ☐ Mouse only
    - ☐ Keyboard only
    - ☐ Mouse and Keyboard

(C) Website

12. Other than games, what functions should be included in the website?
    - ☐ Notes
    - ☐ Sample source codes
    - ☐ Programming exercises
    - ☐ Links to other relevant sites
    - ☐ Others (Please specify: _____)

13. Should the website be assigned access levels (eg. member and non-member. Only member has the privilege to access the content of the website)?

- ☐ Yes
- ☐ No

This is the end of the questionnaire. Thank you.

APPENDIX B

INTERVIEW QUESTIONS

Q.1 What do you think of the scope of the games? Is it sufficient to learn C++?

Q.2 Is the design attractive? Does it make learners more interested in playing the game and learning at the same time? Is it suitable for learners at university level?

Q.3 What do you think of the content of the games? Are the questions in the games appropriate?

Q.4 Are the games effective in learning C++?

Q.5 What do you think of the website? Is the design attractive? How about the colour combination?

Q.6 Is the website easy to navigate and use?

Q.7 Please give comments and/or suggestions on the system

USER MANUAL / GUIDE

Table of Content

List of Figures

CHAPTER 1: INTRODUCTION

Digital game-based learning system: C++, is about learning procedural programming in C++. It aims at helping learners of C++ programming language, developing the motivation to be more interested in learning C++ and enhancing their understanding of procedural programming by playing games and accessing contents provided by the website.

Procedural programming in C++ encompasses basic programming theory, followed by control structures, functions, arrays and lastly, pointers.

CHAPTER 2: GETTING STARTED

Before you can use the system you must ensure you have all the basic software, hardware installed in your computer and ensure that it fulfils the minimum requirements required to run this system.

2.1 System Requirements

2.1.1 Hardware Requirements

These are the hardware requirements

- PC with at least Pentium III pro
- 128MB RAM
- 8GB of Hard Disk Memory
- 15" monitor, 16-bit colour screen
- Keyboard
- Mouse

- Modem

2.1.2 Software Requirements

The software requirements are:

- Windows 98 and above
- Internet Explorer 6.0
- Macromedia Flash 7.0 plug-in

2.1.3 Acquiring Macromedia Flash 7.0 plug-in

Most computers do not come with Macromedia Flash's plug-in. It is a must that Flash plug-in is available in your computer to play the games provided in the website. The flash plug-in that is required to use the contents of the website is the newest version – Flash 7.0 plug-in. You can download the Macromedia Flash 7.0 plug-in from www.macromedia.com/software/flash. It will only take a few minutes depending on your connection speed.

CHAPTER 3: HOW TO USE THE WEBSITE

3.1 Registration

Before you may access the content of the website, user must register into the system.

Steps:

1. Click on the "New User?" link on the Login Page (Figure 1).

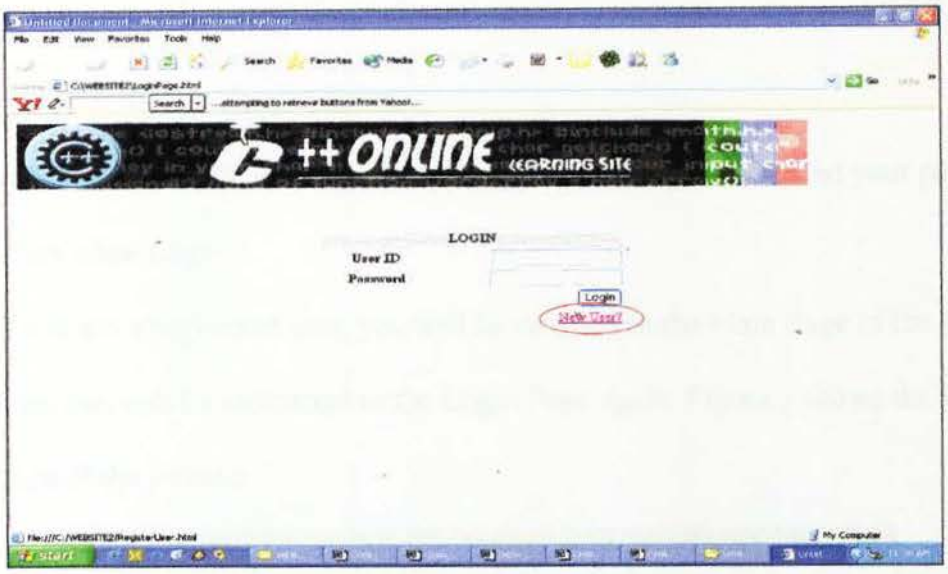


Figure 1: Click on "New User?" to register

2. Then, fill in the required details [name = userID, password and email] in the Registration Page

3. Click Submit upon completion (Figure 2). Your details will be stored into a database. You may now proceed to login to the website.

Figure 2: Registration Page

3.2 Login

To login,

1. Fill in your userID, which is your "name" during registration and your password.
2. Then, click Login.
3. If you are a registered user, you will be directed to the Main Page of the website.
4. Else, you will be redirected to the Login Page again. Figure 3 shows the Main page of the website

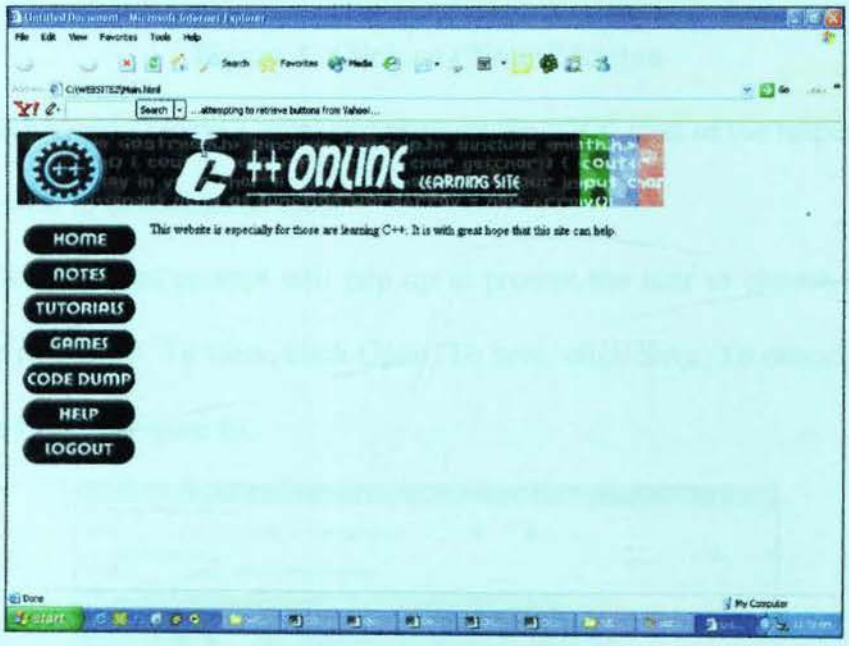


Figure 3: Main Page

3.3 Downloading the content of the website

3.3.1 Downloading notes

To download notes,

1. Click Notes button in the Main page (Figure 4). You will be directed to the Notes Page (Figure 5).

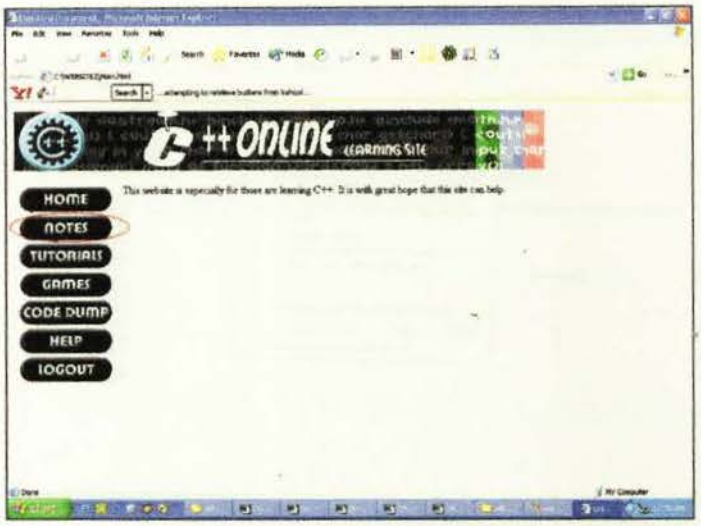


Figure 4: Click on "Notes" button

2. To download or view a note, just click on the "GO" icon of the respective note you wish to download or view.

3. A file download prompt will pop up to prompt the user to choose either Open, Save or Cancel. To view, click Open. To save, click Save. To cancel the process, click Cancel (Figure 6).

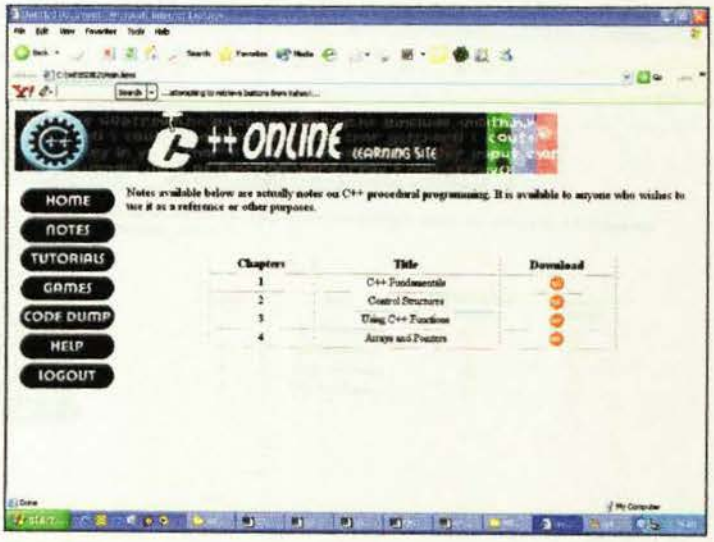


Figure 5: Notes Page

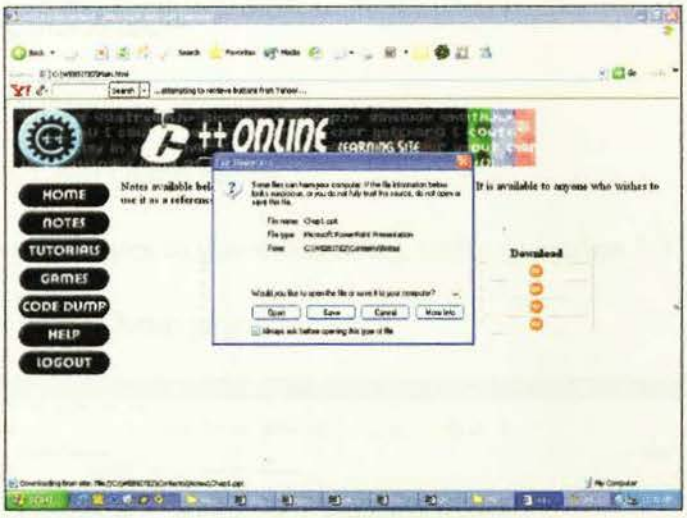


Figure 6: Prompt for file download

3.3.2 Downloading tutorials

The steps to download tutorials and lab questions are similar to downloading notes in the Notes page. Please refer to section 3.3.2. Figure 7 shows the Tutorials Page.

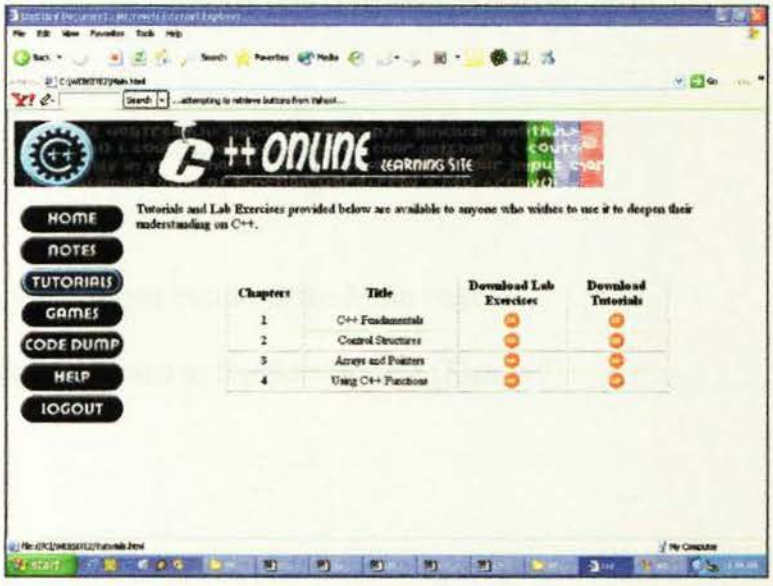


Figure 7: Tutorials Page

3.3.3 Downloading source codes

To download source codes,

1. Click on the Code Dump button.

2. Follow the procedures to download notes. (refer to section 3.3.1).

Figure 8 shows the Code Dump page.

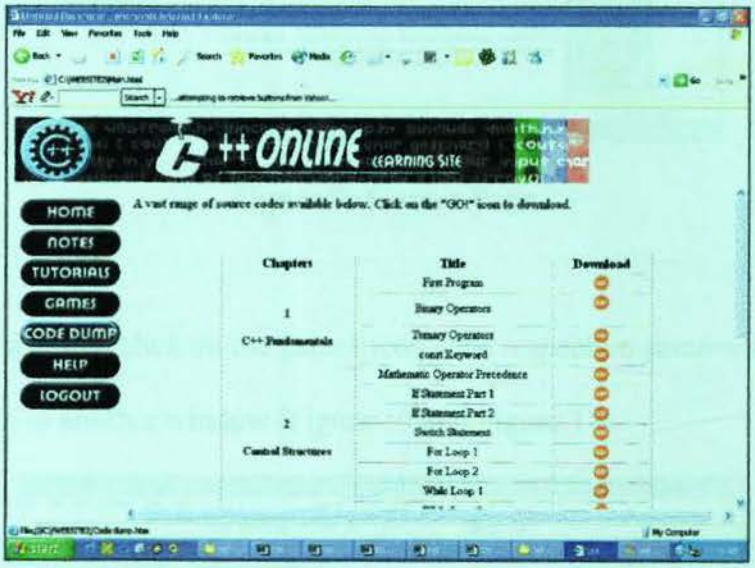


Figure 8: Code Dump Page

3.4 Downloading and Playing Games

Steps:

1. Click on the Games button in the Main Page

2. You will be directed to the Games Page (Figure 9)

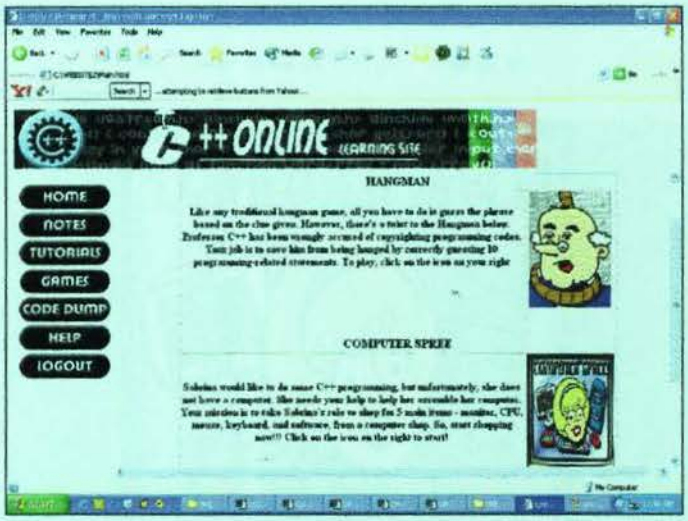


Figure 9: Games Page

3. To play a game, click on the games icon. The respective game will be made available in another window (Figure 10 and Figure 11).

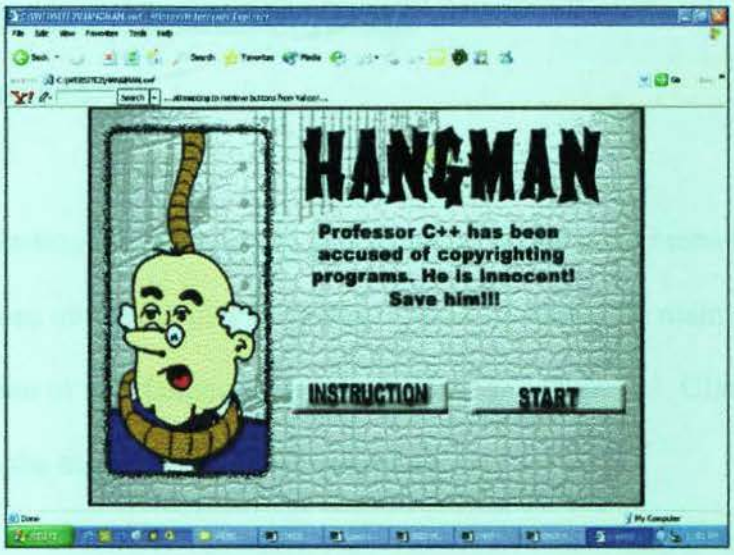


Figure 10: Hangman Game

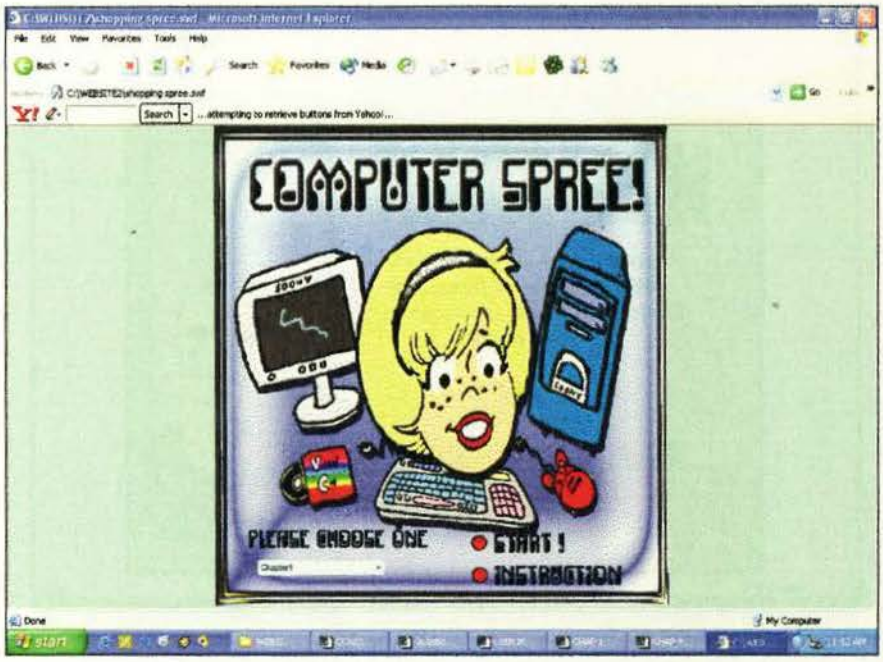


Figure 11: Computer Spree Game

CHAPTER 4: HOW TO PLAY THE GAME

4.1 Playing Hangman

Steps:

1. Before starting the game, click on Instruction on the main menu to view instructions on how to play the game. Figure 12 shows the main menu. Instruction of the Hangman game is displayed in Figure 13. Click on Back to return to the main menu of the Hangman game.

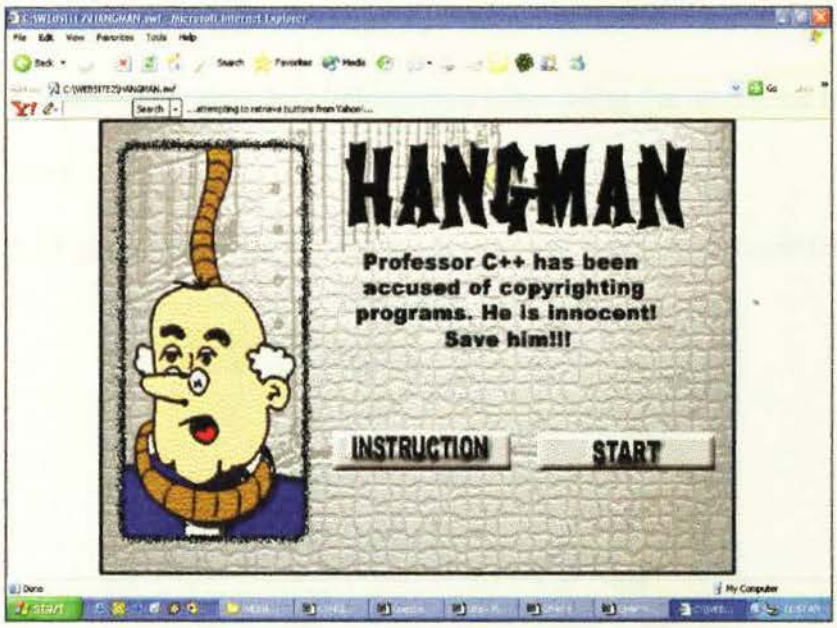


Figure 12: Main menu of Hangman Game

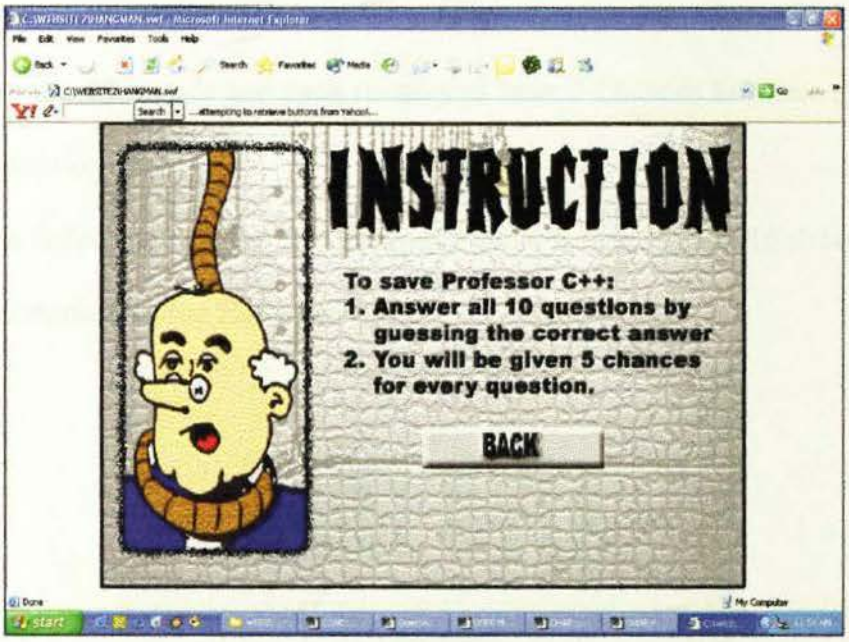


Figure 13: Hangman Instruction

2. Click on Start on the main menu to start the game.

3. Question and empty guessing slots can be seen once you start the game. Guess letters in the empty slots by clicking on a letter using the mouse on the line of characters.

Figure 14 shows the question, guessing slots and the line of character.

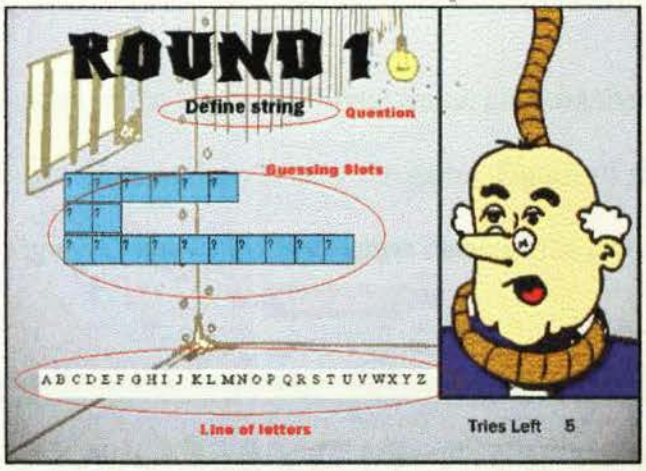


Figure 14: The contents of the Hangman game

4. There are 10 rounds and each round you have 5 chances to guess the answer to the question asked.

5. If you failed to finish the 10 rounds, the game is over. Figure 15 shows the game over interface of the Hangman game

Figure 15: Game over interface of Hangman game

6. If you succeed, you have won the game.

4.2 Playing Computer Spree

Steps:

1. On the main menu of the Computer Spree game, click on the Instruction button to view the instructions on how to play the game. Figure 16 shows the main menu while Figure 17 shows the instruction menu of the game.

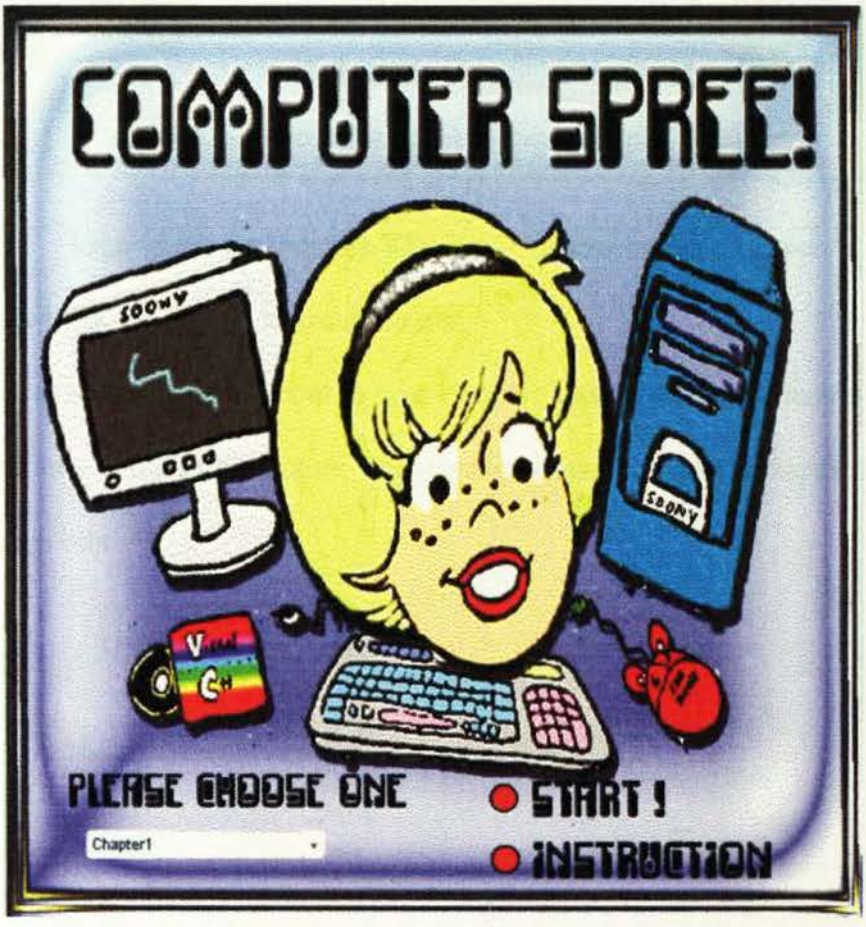


Figure 16: Main Menu of Computer Spree game

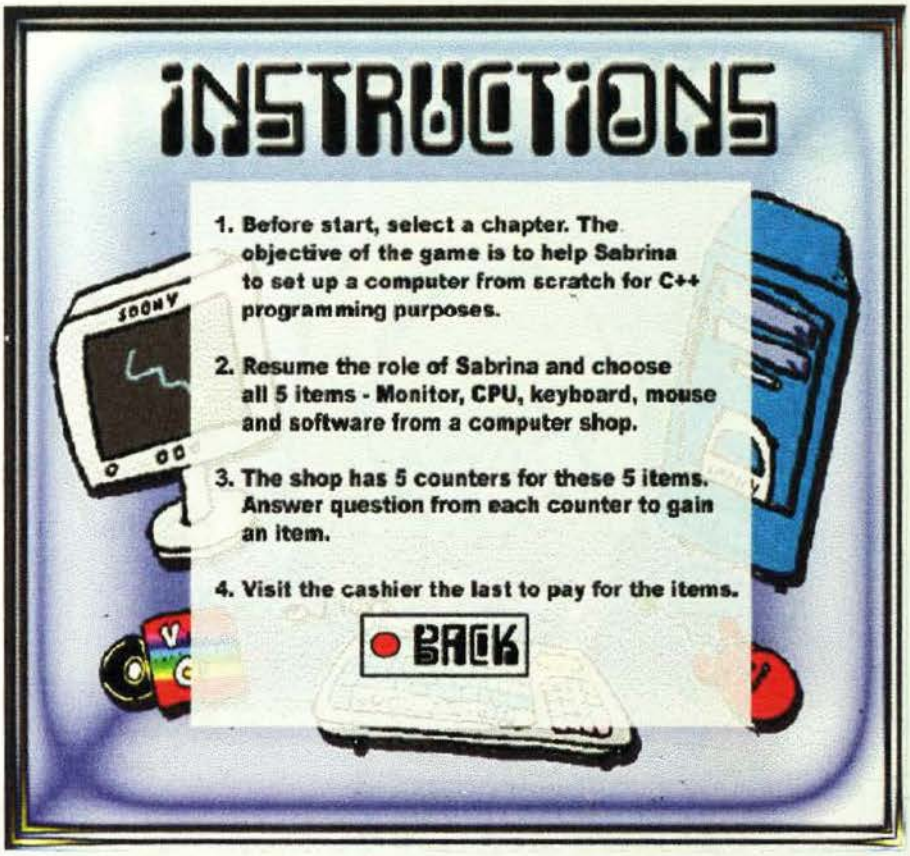


Figure 17: Instruction Menu of Computer Spree game

2. Click Back to go back to the main menu

3. Choose a chapter in the combo box. Chapter chosen will be reflected on the questions in the game. Figure 18 shows the how to choose a chapter in the combo box

Figure 18: Choosing a chapter in the combo box

4. Click on start to start the game
5. Use your keyboard arrow keys to navigate "Sabrina" through the counters.
6. Upon arriving at a counter, questions will be asked. Click using your mouse to select an answer. Figure 19 shows the questions and answer selections in the game.

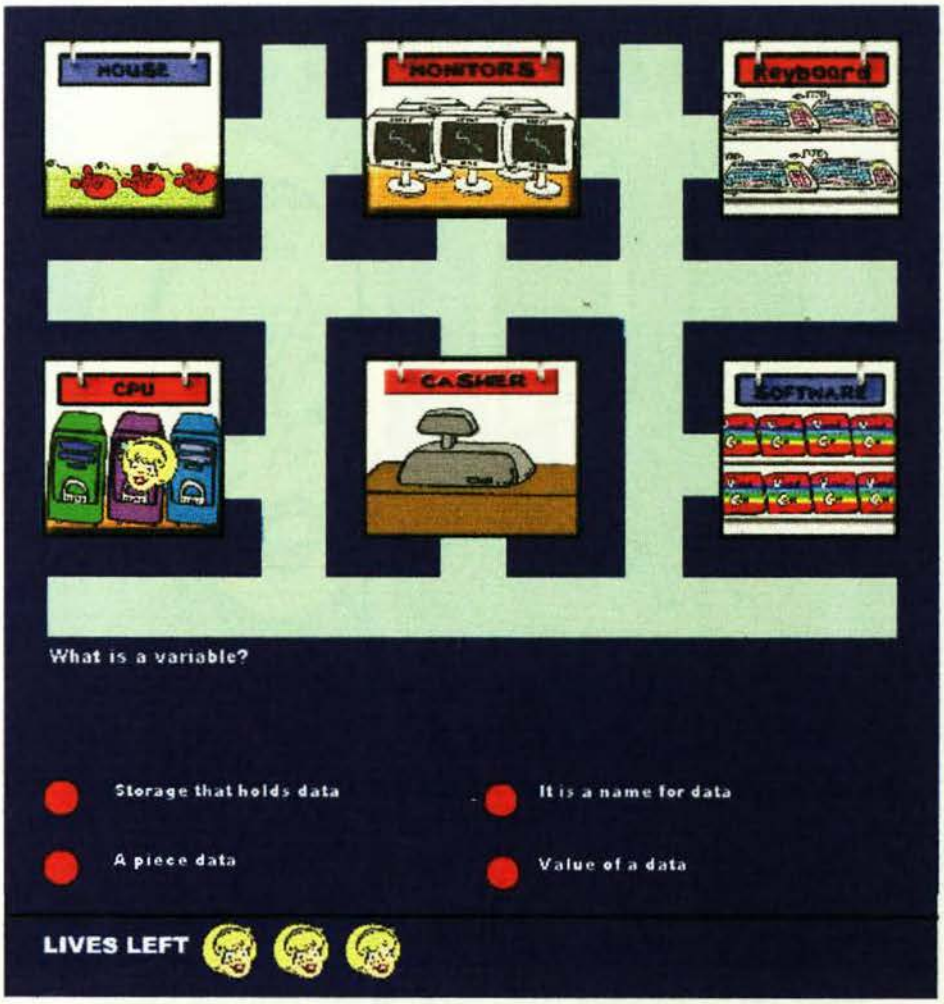


Figure 19: Questions and answers options in the game

7. If you answer all questions correctly, you won the game. You may restart the game by clicking on Play Again. Figure 20 shows the interface upon winning the game.

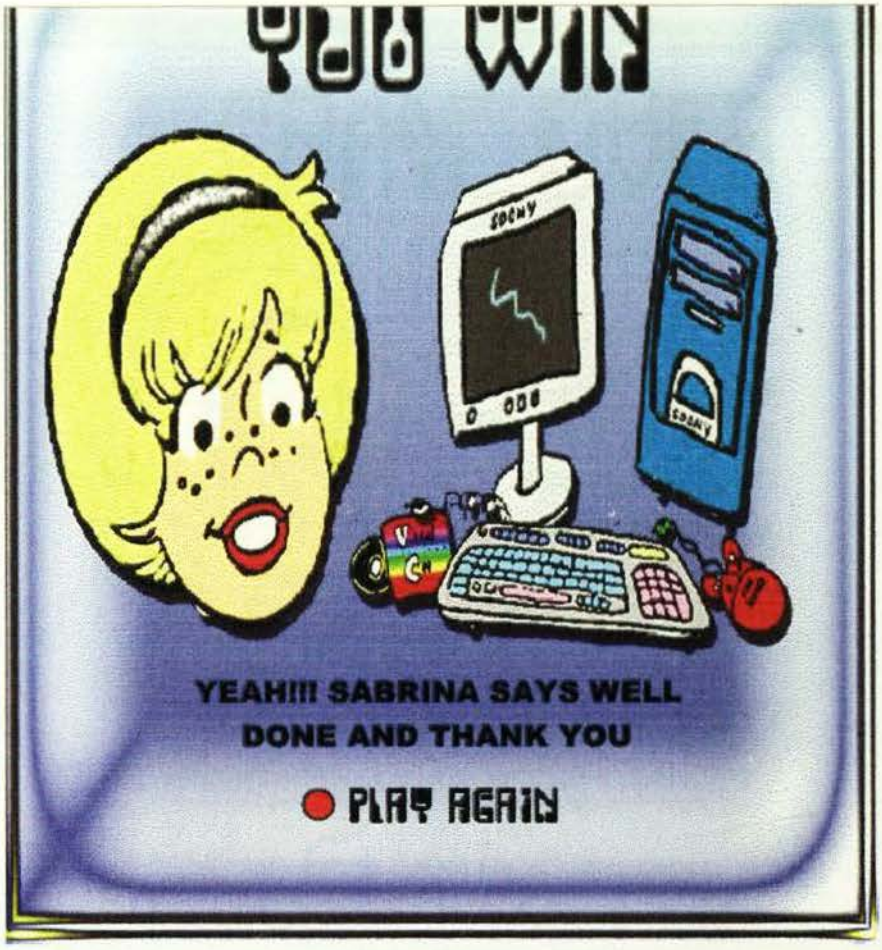


Figure 20: Interface showing the completion of the game

8. If you used up all your lives, you lost the game. You will be directed to the game over menu. Figure 21 shows the game over interface of the game

Figure 21: Game over interface of the game